Corpus-Based Methods in Chinese Morphology and Phonology

中文構詞法和音韻學語料庫方法

Richard Sproat (史伯樂) AT&T Labs – Research
Chilin Shih (石基琳) Bell Labs

Contents

0 Overview

These notes were developed for a course that was presented at the 2001 Summer Institute of the Linguistic Society of America, in the Subinstitute on Chinese Corpus Linguistics at the University of California, Santa Barbara, July 15 – August 3, 2001.

The first section is an introduction to Chinese morphology with a particular focus on the question of what is a word. The second section introduces three topics: properties of word frequency distributions, measures of productivity, and measures of association. The third sectionwill discuss some applications to linguistic and computational linguistic problems. Finally, the last section deals with corpus-based methods in Mandarin phonology.

We would like to thank Martin Jansche for useful feedback on a portion of these notes, and Harald Baayen for providing us with a prepublication copy of his book. We also thank Academia Sinica for generously allowing us to use their text corpora for the purposes of this course and Chu-Ren Huang for his role in negotiating this use.

1 Morphology

1.1 What is a word in Chinese?

Before one can embark upon a discussion of morphology for any language, it is usually a good idea to have some idea of what the scope of the topic is. Basically, what is a word and how do you know when you've got one? In Chinese, this question is a particularly difficult one due to the well-known fact that Chinese orthography systematically fails to represent word boundaries.

As we shall see, the definition of word in Chinese is by no means clear: later on (Section 1.3) we will discuss three proposed standards for segmenting Chinese text into words, and it will be seen that these standards differ in how they classify units as either single or multiple words. Theoretical linguistic definitions, such as the one proposed by Packard (2000), which we'll discuss momentarily, tend to be more principled at an abstract level, but harder to pin down when it comes to specifics.

One thing that most people would certainly agree on is that Chinese words (as words in any language) are constructed out of one or more morphemes. So a good starting point for a discussion of Chinese morphology is a definition of the morpheme. At a first glance, one might be tempted to think that this, at least, has a simple answer: a morpheme in Chinese is something that is written with a single character, and pronounced as a single syllable. By and large, this actually seems to be true, but there are still a lot of problems around the edges. The major problem is that there seem to be a lot of disyllabic morphemes; if one extends the discussion to foreign names that have been borrowed into Chinese, such as 巴基斯坦 *bājīsītǎn* 'Pakistan', then morphemes can be even longer than disyllabic. For native forms it is often not so easy to say whether they are one morpheme or two. A familiar case is 東西 *dōngxī* (east west) 'things', which presumably does not literally derive from a compound meaning 'east and west'. Still the semantic relation between 'east and west' (as in '(all things) between east and west') and 'things' is about as close as the semantic relation between 'suddenly' and 'on a horse'. Yet 馬上 *mǎshàng* (horse on) 'suddenly' is commonly assumed to derive from its literal source.

One clear set of disyllabic morphemes are cases where both characters are written with the same semantic radical. Some examples of these are given in Table 1, taken from (Sproat, 2000). These data also illustrate the (simple) first application discussed in this course of a corpus-based method in morphology: the examples cited were all collected from a 20 million character corpus, and consist of pairs of characters that cooccur only with each other, where the pair itself occurs more than twice.

Having a slightly clearer idea of what a morpheme in Chinese is, we still need to understand what a word is. The most complete recent discussion of this topic is that of Packard (2000), where he discusses the following notions of word:

- **Orthographic word:** Words as defined by delimiters in written text. This appears to have no relevance in Chinese since there are no such written delimiters (apart from punctuation marks, which mark ends of phrases, not words).

Orthography	Analysis	Pronunciation	Gloss
蹉跎	<**foot**+**cuōtuō** >	*cuōtuó*	‘procrastinate’
踉蹌	<**foot**+**liángcāng** >	*lángqiāng*	‘hobble’
蹂躪	<**foot**+**róulìn** >	*róulìn*	‘trample’
躊躇	<**foot**+**shòuzhù** >	*chóuchú*	‘hesitate’
躑躅	<**foot**+**zhèngshǔ** >	*zhízhú*	‘hesitate’
囹圄	<**surround**+**lìngwú** >	*língyú*	‘imprisoned’
囫圇	<**surround**+**wùlún** >	*húlún*	‘swallow whole’
轇轕	<**cart**+**liàogě** >	*jiūgé*	‘entwined’
魍魎	<**demon**+**wǎngliǎng** >	*wǎngliǎng*	‘roaming ghost’
妯娌	<**female**+**yóulǐ** >	*zhóulǐ*	‘sister in laws’
餛飩	<**food**+**kūntún** >	*húntún*	‘wonton’
荸薺	<**grass**+**bóqí** >	*bíqí*	‘water chestnut’
萵苣	<**grass**+**guǎjù** >	*wōjù*	‘lettuce’
菡萏	<**grass**+**hánxiàn** >	*hàndàn*	‘lotus’
蒹葭	<**grass**+**jiānjiǎ** >	*jiānjiā*	‘type of reed’
苜蓿	<**grass**+**mùsù** >	*mùsù*	‘clover’
窈窕	<**cave**+**yòutiao** >	*yǎotiǎo*	‘graceful’
迤邐	<**going**+**yílǐ** >	*yǐlǐ*	‘trailing’
揶揄	<**hand**+**yēyú** >	*yéyú*	‘tease’
顢頇	<**head**+**mǎnhan**>	*mánhān*	‘muddleheaded’
慫恿	<**heart**+**cóngyǒng** >	*sǒngyǒng*	‘egg on’
忸怩	<**heart**+**niuní** >	*niǔní*	‘coy’
慇懃	<**heart**+**yīnqín** >	*yīnqín*	‘attentively’
蝙蝠	<**insect**+**biǎnfù** >	*biānfú*	‘bat’
蜉蝣	<**insect**+**fúyóu** >	*fúyóu*	‘mayfly’
蚯蚓	<**insect**+**qiūyǐn** >	*qiūyǐn*	‘earthworm’
氤氳	<**gas**+**yīnyun** >	*yīnyūn*	‘misty atmosphere’
邂逅	<**going**+**xièhòu** >	*xièhòu*	‘encounter’
璀璨	<**jade**+**cuīcàn** >	*cuǐcàn*	‘brilliant’
玳瑁	<**jade**+**dàimào** >	*dàimào*	‘tortoise shell’
鞦韆	<**leather**+**qiūqiān** >	*qiūqiān*	‘swing’
耄耋	<**old**+**máozhì** >	*màodié*	‘old people’
旖旎	<**overhanging**+**yiní** >	*yǐnǐ*	‘fluttering’
倥傯	<**person**+**kōngzǒng**>	*kǒngzǒng*	‘busy’
疙瘩	<**sickness**+**gedá** >	*gēdā*	‘cyst, boil’
齟齬	<**teeth**+**jūwú** >	*jǔyǔ*	‘bickering’
枇杷	<**tree**+**pibā** >	*pípá*	‘loquat’
檸檬	<**tree**+**níngméng** >	*níngméng*	‘lemon’

Table 1: Pairs of characters that only cooccur with each other, and occur at least three times in a 20 million character corpus. Note that in each case both characters share the same semantic radical. See, for instance, the examples with the **foot** radical, underlined in the table above. The second column gives the component analysis following the schema in (Sproat, 2000).

- **Sociological word:** Following Chao (1968, pp. 136–138), these are 'that type of unit, intermediate in size between a phoneme and a sentence, which the general, non-linguistic public is conscious of, talks about, has an everyday term for, and is practically concerned with in various ways.' In English this is the lay notion of 'word', whereas in Chinese this is the character (字 *zì*).

- **Lexical word:** This corresponds to Di Sciullo and Williams' (Di Sciullo and Williams, 1987) *listeme*.

- **Semantic word:** Roughly speaking, this corresponds to a 'unitary concept'.

- **Phonological word:** A word-sized entity that is defined using phonological criteria. (Yes, that's circular.) Packard goes on to note that Chao considers prosodic issues, such as when one can pause in a sentence, to provide useful tests for phonological wordhood. In many languages, phonological words are the domain of particular phonological processes: in Finnish, for example, vowel harmony is restricted to phonological words. In dialects of Mandarin that have (stress) reduction, such phenomena are generally restricted to words, as is consonant lenition (see the section on Phonology).

- **Morphological word:** following Di Sciullo and Williams' notion, a morphological word is anything that is the output of a word-formation rule.

- **Syntactic word:** These are all and only those constructions that can occupy X^0 slots in the syntax. (The syntactic word is the definition of word that Packard uses as the basis for the bulk of his discussion.)

- **Psycholinguistic word:** This is the "'word' level of linguistic analysis that is . . . salient and highly relevant to the operation of the language processor" (Packard, 2000, p. 13).

To these definitions could be added the extrinsically defined notions of word that have formed the basis of the various segmentation standards for Chinese that have been proposed, and that we shall discuss in Section 1.3. It is likely that none of these notions of word will completely line up with any other definition. As a practical matter this probably doesn't matter. After all, the definition of word that is most useful will depend to a large degree upon what one wants to use it for. In Chinese language technology, a phonological word is likely to be of interest if one is interested in an application such as text-to-speech synthesis; if one is interested in machine translation, then it is likely that one would be looking more at semantic words; and if one is constructing lexicons, then some notion of lexical word will be relevant. It really doesn't matter that these are different. Note that proposals for segmentation standards, such as those described in Section 1.3 largely gloss over these considerations and assume that there is a single "correct" segmentation.

1.2 Some Chinese Morphological Phenomena

In this section we give an overview of various kinds of morphological phenomena in Chinese. Many of the examples are culled from Li and Thompson (1981) and Packard (2000).

In addition to reduplication, affixation and compounding — the types of categories that one would normally expect to see discussed in a treatment of morphology — we also discuss names, and the kind of Chinese abbreviations known as 縮寫 *suōxiě* or 'shrunken writing'. The reason for including these is that in any practical application involving words in real text, one will see plenty of examples of these categories, so their formal properties are worth considering. Note that among theoretical treatments of Chinese morphology, at least Packard discusses *suoxie* at some length.

1.2.1 Verbal reduplication

Verbal reduplication is fairly productive in Mandarin, and is typically used to convey the meaning 'X a little', where 'X' is the meaning of the verb. The form of the reduplication is either a simple copy of the verb, or else if the verb is monosyllabic, 一 *yī* 'one' may be used between the copies. Some examples of each of these forms are given in (1) and (2):

(1)	說說	shuō-shuō	(speak speak)	'speak a little'
	看看	kàn-kàn	(look look)	'have a little look'
	走走	zǒu-zǒu	(walk walk)	'have a little walk'
	磨磨	mó-mó	(rub rub)	'rub a little'
	討論討論	tǎolùn-tǎolùn	(discuss discuss)	'discuss a little'
	請教請教	qǐngjiào-qǐngjiào	(ask ask)	'ask a little'
	研究研究	yánjiù-yánjiù	(research research)	'research into'

(2)	說一說	shuō-yi-shuō	(say one say)	'go ahead and say it'
	看一看	kàn-yi-kàn	(look one look)	'have a look'
	走一走	zǒu-yi-zǒu	(walk one walk)	'have a little walk'
	磨一磨	mó-yi-mó	(rub one rub)	'rub a little'

Reduplication can also cooccur with 看 *kàn* 'see' with the rough meaning 'do X and see how it goes':

(3)	說說看	shuō-shuō-kàn	(speak speak look)	'talk about it and see'
	走走看	zǒu-zǒu-kàn	(walk walk look)	'walk and see'

1.2.2 Adjectival Reduplication

Like verbs, adjectives can reduplicate in Mandarin. These reduplicated forms typically have the meaning 'a bit X':

(4) 紅紅 *hóng-hóng* (red red) ‘red’
慢慢 *màn-màn* (slow slow) ‘slow’

Disyllabic adjectives reduplicate in the pattern AABB:

(5) 舒服 *shūfú* 舒舒服服 *shūshū-fúfú* ‘comfortable’
乾淨 *gānjìng* 乾乾淨淨 *gāngān-jìngjìng* ‘clean’
胡塗 *hútú* 胡胡塗塗 *húhú-tútú* ‘muddle-headed’
快樂 *kuàilè* 快快樂樂 *kuàikuài-lèlè* ‘happy’
漂亮 *piàoliàng* 漂漂亮亮 *piàopiào-liàngliàng* ‘pretty’

One also finds the pattern ABB:

(6) 亮晶 *liàngjīng* 亮晶晶 *liàng-jīngjīng* ‘bright’
白花 *báihuā* 白花花 *bái-huāhuā* ‘white’

There are various semantic and stylistic restrictions on adjectival reduplication. Thus the following are all unacceptable:

(7) ∗重重要要 *zhòngzhòng-yàoyào* ‘important’
∗偉偉大大 *wěiwěi-dàdà* ‘majestic’
∗粉粉紅紅 *fēnfēn-hónghóng* ‘pink’
∗?美美麗麗 *měiměi-lìlì* ‘beautiful’
∗透透明明 *tòutòu-míngmíng* ‘transparent’

In some cases there seems to be a semantic clash. This is clear in the case of 偉大 *wěidà* ‘majestic’, where even in English the phrase ?*a bit majestic* seems odd. In the case of 賢明 ‘sagacious’, the adjective is apparently too classical to admit of reduplication. In other cases the restriction is less clear.

1.2.3 Measure word reduplication

Measure words can also reduplicate in Mandarin, typically with the meaning of ‘every X’: thus 磅 *bàng* ‘pound’ yields 磅磅 *bàngbàng* ‘every pound’. Some examples follow:

(8) 磅磅 *bàngbàng* ‘every pound’
條條魚 *tiáotiáo yú* ‘every fish’
套套西裝 *tàotào xīzhuāng* ‘every suit’
件件衣服 *jiànjiàn yīfú* ‘every piece of clothing’

Not all measure words seem to allow this reduplication. Thus:

(9)

∗雙雙手	*zhīzhī shǒu*	‘every hand’
∗冊冊書	*cècè shū*	‘every book’
∗打打蛋	*dádá dàn*	‘every dozen eggs’
?座座山	*zuòzuò shān*	‘every mountain’
?隻隻雞	*zhīzhī jī*	‘every chicken’

Note however in contrast to the final example in (9), where the duplicated measure word is adjacent to the noun; the construction is fine if the duplicated form is separated from the noun, as in (10):

(10) 我們的雞,隻隻都病了
wǒmen-de jī, zhīzhī dōu bìng le
(our-DE chicken CL-CL all sick ASP)
‘Every one of our chickens is sick.’

In some cases similar examples appear to behave differently. Thus ?日日 *rìrì* ‘every day’ is less acceptable than 天天 *tiāntiān* ‘every day’.

(11)

?日日	*rìrì*	‘every day’ (okay in poetic language)
天天	*tiāntiān*	‘every day’
年年	*niánnián*	‘every year’

Finally, measure word reduplication appears to be limited to monosyllabic measure words:

(12) ∗公里公里 *gōnglǐ gōnglǐ* ‘every kilometer’

1.2.4 Prefixation

Mandarin has a few productive or semi-productive prefixes. These include: 老 *lǎo-* used with proper names; 小 *xiǎo-* ‘little’, also used with names; the ordinal prefix 第 *dì*; the calendrical prefix 初 *chū*; and the verbal prefixes 好 *hǎo-* ‘good’ and 難 *nán-* ‘bad’. These are exemplified in (13):

(13)

老	*lǎo-*	老王	*lǎo wáng*	‘old Wang’
小	*xiǎo-*	小張	*xiǎo zhāng*	‘little Zhang’
第	*dì-*	第一	*dìyī*	‘first’
初	*chū-*	初三	*chū sān*	‘the third (day of the month)’
好/難	*hǎo-/nán-*	好吃/難吃	*hǎochī/nánchī*	‘tasty/bad-tasting’

Another prefix is 可 *kě-* meaning ‘-able’. Thus:

(14)

可	*kě-*	可愛	*kě-ài*	‘lovable, cute’
		可靠	*kě-kào*	‘reliable’

This is claimed — e.g. by Li and Thompson (1981) — not to be very productive.

Finally, a clearly non-productive prefix is 大 *dà-* ‘big’, which is used in a few nouns, such as 大象 *dàxiàng* ‘elephant’.

1.2.5 Suffixation

Productive suffixes are more plentiful in Mandarin than productive prefixes. Following, most recently, Packard (2000), we can classify suffixes into two categories, namely derivational and inflectional. Examples of derivational suffixes are:

(15) "Diminutive" suffixes:

兒	*-er*	鳥兒	*niǎo-er*	'bird'
子	*-zi*	猴子	*hóu-zi*	'monkey'
頭	*-tou*	饅頭	*mán-tou*	'steamed bread'

(16) Other deriational suffixes:

學	*-xué*	心理學	*xīnlǐ-xué*	'psychology'
家	*-jiā*	物理學家	*wùlǐxué-jiā*	'physicist'
化	*-huà*	美化	*měi-huà*	'Americanization'
率	*-l`ü*	生蛋率	*shēng-dàn-l`ü*	'egg production rate'
主義	*-zhǔyì*	馬克斯主義	*mǎkèsī-zhǔyì*	'Marxism'

Common inflectional suffixes include aspectual markers like 了 *-le* and 過 *-guò*, and the human noun plural 們 *-men*:

(17)

了	*-le*	吃了	*chī-le*	'have eaten'
過	*-guò*	吃過了	*chī-guò-le*	'have eaten'
們	*-men*	孩子們	*háizi-men*	'children'

Mandarin also has a wide range of resultative markers which might be considered either instances of suffixation or instances of compounding. These are discussed below.

Inflectional suffixes are one category where the different segmentation schemes discussed in Section 1.3 differ: thus the Mainland scheme and (presumably) the ROCLING scheme would segment 朋友們 *péngyǒumen* 'friends' as two words (though 我們 *wǒmén* 'we' would be two words), whereas the University of Pennsylvania Chinese Treebank scheme would segment it as one word.

1.2.6 Compounding

Compounding comprises the largest category of morphological derivations in Mandarin. The cases we will consider here are nominal compounding and verbal compounding. There is also a category of adjectival (or, if you prefer, stative verbal), such as 粉紅 *fěnhóng* (powder red) 'pink' or 雪白 *xuěbái* 'snow white'.

Among nominal compounds one can distinguish full-word compounding, where the components are each full words, and what elsewhere (Sproat and Shih, 1996) we have termed *root compounds*, where at least one of the elements is not a free-standing word; see also (Packard, 2000).

Verbal compounding includes resultatives, "parallel" (V-V) compounds, subject-verb compounds and verb-object compounds. In principle, as Packard shows, each of these can either

involve full words or roots, though we'll mostly concentrate here on the cases involving full words.

Each of these compound types are exemplified in the following sections.

1.2.6.1 Full-Word Nominal Compounding. Noun compounds in Mandarin display the same range of meanings as equivalent constructions in English. For example, a compound XY can have a **locational** reading ('a Y that is located in X'), a **use** reading ('a Y that is used for X') or a **material** reading ('a Y that is made out of X'), among others. Some examples of several kinds are illustrated below:

(18) **location**

客廳 沙發	*kètīng shāfā*	'living room sofa'
河 馬	*hémǎ*	'hippopotamus (river horse)'
海 狗	*hǎigǒu*	'seal (sea dog)'

(19) **used for**

指甲油	*zhǐjiǎ yóu*	'nail polish'
乒乓球	*pīngpāng qiú*	'pingpong ball'
太陽眼鏡	*tàiyáng yǎnjìng*	'sunglasses'
飯碗	*fànwǎn*	'rice bowl'

(20) **material**

大理石 地板	*dàlǐshídìbǎn*	'marble (Dali rock) floor'
紙老虎	*zhǐlǎohǔ*	'paper tiger'

(21) **powered by**

電 燈	*diàn dēng*	'electric light'
風 車	*fēng chē*	'windmill (wind cart)'

(22) **organization**

大學 校長	*dàxué xiàozhǎng*	'university president'
北京 大學	*běijīng dàxué*	'Beijing University'

(北京大學 might also be considered to be an instance of **location**). The only relatively productive type of compounds in Mandarin that have no productive counterpart in English are **coordinate** (often also called *dvandva*) compounds. These are illustrated below:

(23) **coordinate**

爸爸 媽媽	*bàbà māmā*	'father and mother'
風水	*fēng shuǐ*	'fengshui (wind water)'
紙筆	*zhǐ bǐ*	'paper and pen'
牛羊	*niú yáng*	'livestock (cattle sheep)'

1.2.6.2 Root Nominal Compounding. Chinese has a large number of what we will term *root compounding* following earlier work (Sproat and Shih, 1996); Packard (2000) calls these *bound root words*. An example is the word for 'termite' 白蟻 *báiyǐ*, literally 'white ant', where we underline the portion meaning 'ant'. The normal Mandarin word for 'ant' is 螞蟻 *mǎyǐ*, and in normal Mandarin discourse, 蟻 *yǐ* cannot be used as a separate word. One does find examples like, where the first 蟻 *yǐ* and 蜂 *fēng* in each clause are apparently functioning as free words:

(24) 蟻有蟻國,蜂有蜂國
yǐ yǒu yǐguó, fēng yǒu fēngguó
(ant have ant country, bee have bee country)
'Ants have Antland, bees have Beeland'

But this is clearly a case of intentional classical style: as Packard notes, apparent cases of fluidity in what counts as a word in Chinese, nearly always are due to influences of register and style.

Some examples of root compounding follow. In the first column of each of the examples, the normal Mandarin word is given with the root underlined. In the second and third columns examples are given with the root in question in the head or non-head position of the compound.[1] The fact that these roots can occur in both head and non-head position suggests that the process in question has more in common with compounding than it does with affixation, since affixes typically select to attach on one side or the other of their stems, but not both.

(25)	螞蟻 *mǎyǐ* 'ant'	蟻王	工蟻
		yǐwáng 'queen ant'	*gōngyǐ* 'worker ant'
	腦子 *nǎozi* 'brain'	腦水腫	後腦
		nǎoshuǐzhǒng 'hydrocephaly'	*hòunǎo* 'hindbrain'
	蒼蠅 *cāngyíng* 'fly'	蠅屍	地中海蠅
		yíngshī 'fly corpse'	*dìzhōnghǎiyíng* 'Mediterranean fly'
	蘑菇 *mógū* 'mushroom'	菇傘	金菇
		gūsǎn 'pileus'	*jīngū* 'golden mushroom'

Sproat and Shih (1996) argue that root compounding is productive; we'll examine those arguments later on. For now note that that conclusion argues against Dai (1992), who claims that only morphology involving full words is productive.

1.2.6.3 Resultative Compounds. The most productive class of verbal compounds in Mandarin are the resultative compounds. These are formed by the suffixation of a resultative marker to the stem. Some common resultative markers are:

(26) 上 *shàng* 'up'
下 *xià* 'down'

[1] See Packard (2000) for an extensive discussion of headedness in Mandarin.

進 *jìn* ‘enter’
出 *chū* ‘(go) out’
起來 *qǐlái* inchoative
回 *huí* ‘return’
過 *guò* ‘pass, across, beyond’
開 *kāi* ‘open’
完 *wán* ‘finish’
到 *dào* ‘reach’
好 *hǎo* ‘good, complete’
死 *sǐ* ‘die’
見 *jiàn* ‘see, perceive’
破 *pò* ‘break’
清楚 *qīngchǔ* ‘clearly’

The exact meaning of many of these affixes depends upon the verb it combines with. The meanings include **result**, **achievement**, **direction**, among others:

(27)	**result**	打破	*dǎpò*	‘break by hitting’
		拉開	*lākāi*	‘open’
	achievement	寫清楚	*xiěqīngchǔ*	‘write clearly’
		買到	*mǎidào*	‘succeed in buying’
	direction	跳過去	*tiàoguòqù*	‘jump across’
		走進來	*zǒujìnlái*	‘come walking in’

Resultatives can usually be ‘augmented’ with the so-called ‘infixes’ 得 *dé* ‘able’ and 不 *bù* ‘not able’. These add the sense of ‘(not) able to achieve X’ to the resultative. So from 跳過去 *tiàoguòqù* ‘jump across’ one can form:

(28)	跳得過去	*tiàodéguòqù*	‘able to jump across’
	跳不過去	*tiàobùguòqù*	‘not able to jump across’

1.2.6.4 Parallel Compounds. Parallel compounds consist of verbs formed from a pair of other verbs, with a coordinate interpretation of the pair. Some examples:

(29)	購買	*gòu-mǎi*	(buy-buy)	‘buy’
	建築	*jiàn-zhù*	(build-build)	‘build’
	檢查	*jiǎn-chá*	(examine-examine)	‘examine’
	治療	*zhì-liáo*	(treat-treat)	‘treat (a sickness)’

1.2.6.5 Subject-Predicate Compounds. Subject-predicate compounds consist of a noun followed by a verb for which the noun acts as a subject. For example:

(30)	頭疼	*tóu-téng*	(head hurt)	‘(have a) headache’
	嘴硬	*zuǐ-yìng*	(mouth hard)	‘stubborn’
	眼紅	*yǎn-hóng*	(eye red)	‘covet’
	心酸	*xīn-suān*	(heart sour)	‘feel sad’
	命苦	*mìng-kǔ*	(life bitter)	‘tough straits’

While the noun appears to be assigned a thematic role by the verb, the whole verb itself still assigns an external thematic role: often the thematic role assigned by the subject-predicate verb is notionally the “possessor” of the incorporated noun:

(31) 我頭疼.
wǒ tóu téng
(I head hurt)
‘I have a head ache.’ (= ‘My head hurts.’)

(32) 你真嘴硬!
nǐ zhēn zuǐ-yìng
(you really mouth hard)
‘You are really stubborn!’ (≈ ‘Your mouth is really hard!’)

1.2.6.6 Verb-Object Compounds. A much larger category of verb-argument compounds are verb-object (VO) compounds. Some examples are given in (33):

(33)	出版	*chū-bǎn*	(emit edition)	‘publish’
	睡覺	*shuì-jiào*	(sleep sleep)	‘sleep’
	畢業	*bì-yè*	(finish course of study)	‘graduate’
	開刀	*kāi-dāo*	(operate knife)	‘operate’
	開玩笑	*kāi-wánxiào*	(operate joke)	‘make fun of’
	照像	*zhào-xiàng*	(shine image)	‘take a photo’

The situation with VO’s is more complex than that of subject-predicate compounds. First of all, while some VO compounds allow objects (34), others do not (35):

(34) 他們出版了那本書
tāmen chūbǎn-le nèiběn shū
(they publish-PERF that-CL book)
‘They published that book.’

(35) *他們開刀心臟
tāmen kāidāo xīnzàng
(they operate heart)
'They are operating on the heart'

Second — and this is what has made the structure of VO compounds controversial — the O portion of VO's is generally separable to some degree from the V. This separation can involve marking the V with an aspectual marker (36), marking the N with a numeral-classifier combination (37) and possibly an adjective (38), marking a "possessor" of the object (39), or even preposing (40) or questioning the object (41):

(36) 我睡了覺
wǒ shuì-le jiào
(I sleep-PERF sleep)
'I fell asleep'

(37) 開一個玩笑
kāi yīge wánxiào
(operate one-CL joke)
'make fun of'
照兩張像
zhào liǎngzhāng xiàng
(shine two-CL photo)
'take two photos'

(38) 我睡了一個小覺
wǒ shuì-le yīge xiǎo jiào
(I sleep-PERF one-CL little sleep)
'I took a little nap.'

(39) 開他的玩笑
kāi tāde wánxiào
(operate his joke)
'make fun of him'

(40) 一個便我都沒有大
yīge biàn wǒ dōu méiyǒu dà
(one-CL convenience i all NEG have big)
'I haven't defecated at all.'
(cf. 大便 *dàbiàn* (big convenience) 'defecate')

(41) 你開甚麼刀?
nǐ kāi shénmo dāo

(you operate what knife)
'What kind of operation are you having?'

Various authors, including Chao (1968), J. Huang (1984), and most recently Packard (2000) have presented analyses of VO compounds, focussing on the question of whether these should be considered lexical or phrasal constructions. Packard argues, somewhat along the lines of Huang, that VO compounds may be either words or phrases depending upon the construction one finds them in, but that once a VO construction becomes a word via lexicalization, it is basically a word, and admits of only limited phrasal reanalysis.

On the etymological side, note that a number of VO compounds were not etymologically VO, but have only been reinterpreted as such over time, these include 洗澡 *xǐ-zǎo* 'bathe' and 大便 *dà-biàn* 'defecate'.

1.2.7 Other Kinds of Word Formation.

So far we have been discussing word formation types that would normally be thought of as morphological. But in any Chinese text corpus one will find many instances of words that are formed in ways that one would not normally think of as morphology per se, but which nonetheless exhibit certain regularities. Among these are personal names, foreign words and names in transliteration and abbreviations. Abbreviations, or 縮寫 *suōxiě* (shrunken writing) are the topic of Section 1.2.7.1. We briefly discuss (Chinese) personal names and foreign words in transliteration here.

Full Chinese personal names are in one respect simple: they are always of the form **family+given**. The family name set is restricted: there are a few hundred single-character family names, and about ten double-character ones. Given names are most commonly two characters long, occasionally one character long: there are thus four possible name types, which can be described by a simple set of context-free rewrite rules such as the following:

(42)

1	word	⇒	name
2	name	⇒	1-char-family 2-char-given
3	name	⇒	1-char-family 1-char-given
4	name	⇒	2-char-family 2-char-given
5	name	⇒	2-char-family 1-char-given
6	1-char-family	⇒	char_i
7	2-char-family	⇒	char_i char_j
8	1-char-given	⇒	char_i
9	2-char-given	⇒	char_i char_j

The difficulty is that given names can consist, in principle, of any character or pair of characters, so the possible given names are limited only by the total number of characters, though some characters are certainly far more likely than others.

(The above treatment also fails to cover "double-barreled" family names, which are still commonly used to refer to married women: thus 張王金蘭 *zhāng wáng jīnlán* 'Mrs. Jinlan (Wang) Zhang', though the description could be straightforwardly enough extended to cover these.)

Foreign names are usually transliterated using characters whose sequential pronunciation mimics the source language pronunciation of the name. Since foreign names can be of any length, and since their original pronunciation is effectively unlimited, the identification of such names is tricky. Fortunately, there are only a few hundred characters that are particularly common in transliterations; indeed, the commonest ones, such as 巴 *bā*, 爾 *ěr*, and 阿 *ā* are often clear indicators that a sequence of characters containing them is foreign: even a name like 夏米爾 *xià-mǐ-ěr* 'Shamir', which is a legal Chinese personal name, retains a foreign flavor because of 爾 *ěr*. Some endings are also good indicators of foreign words. Thus for instance, the common English placename 'suffixes', *-nia* (e.g. *Virginia*) and *-sia* (e.g. *Malaysia*) have standard transliterations as 尼亞 *ní-yǎ* and 西亞 *xī-yǎ*, respectively, and words ending in these sequences are invariably transliterated foreign placenames. Also, typographical conventions, such as the use of the center dot '・' are often used to separate the components of foreign names, and these are therefore often cues to a foreign name: 法蘭西斯科・克里蒙 *fǎlánxīsīkē kèlǐméng* 'Francesco Clemente'.

1.2.7.1 Suoxie. *Suoxie* are a common way of producing shortened forms of (especially) long names in Chinese. It is very typical of names of official organizations, but there are quite a few other uses too. While *suoxie* are often also called "abbreviations", they are actually most closely related to acronyms, such as *NATO* or *UNICEF*, in languages such as English. True abbreviations, such as *kg.* in English, are expanded into full words when read: one says *kilogram* rather than *K G*, when reading *kg.*. In contrast, acronyms and *suoxie* are read as words, and not expanded into their original forms.[2]

The formation of *suoxie* is complex, and we will return more fully to the issue in a later section. For now note that *suoxie* are typically formed by taking the *first* character of each component of a word; some of these examples are from (Wang, 1996):

(43) 亞洲乒乓球聯盟 亞乒聯
yǎzhōu pīngpāng qiú liánméng *yǎ pīng lián*
Asian Ping-Pong Association

(44) 工業研究院 工研院
gōngyè yánjiù yuàn *gōng yán yuàn*
Industrial Research Center

(45) 以色列巴勒斯坦和談 以巴和談
yǐsèliè bālèsīdàn hètán *yǐ bā hètán*
Israel-Palestinian discussions

[2]In fact, Chinese seems to lack true abbreviations: even borrowed abbreviations such as *kg.* are read as sequences of letters (*K G*) rather than expanded into the corresponding word. See (Sproat, 2000) for discussion.

As the last example shows, the *suoxie* is not limited to picking morphologically sensible pieces of words: 以色列 *yǐsèliè* 'Israel' and 巴勒斯坦 *bālèsītǎn* 'Palestine', are sound transliterations, and the first syllables used in the *suoxie* example in (45) — 以 *yǐ* and 巴 *bā* respectively — are meaningless components of these transliterations.

Though most *suoxie* involve the first characters of constructions, there are cases where the second character must be used, as in the standard abbreviation of 香港 *xiānggǎng* 'Hong Kong' as 港 *gǎng*:

(46) 台灣香港 台港
táiwān xiānggǎng *tái gǎng*
Taiwan-Hong Kong

中國石油 中油
zhōngguó shíyóu *zhōng yóu*
China Oil

There are also cases where the *suoxie* form is suppletive: this is true for abbreviations of many Chinese Provinces, for example. An instance of this is the form 滬 *hù* for 上海 *shànghǎi* 'Shanghai'.

(47) 上海杭州鐵路 滬杭鐵路
shànghǎi hángzhōu tiělù *hù háng tiělù*
Shanghai-Hangzhou Railway

1.3 Segmentation standards

Deciding what is a word in Chinese is not purely a theoretical or academic exercise: it has practical consequences for a wide variety of applications both "sociological" and technological.

The sociological applications arose early in the twentieth century in the context of various attempts to romanize Chinese orthography: the two competing proposals for romanization were the 國語羅馬字運動 *guǒyǔ luómǎzì yùndòng* 'Mandarin Romanization Movement' (of which Y. R. Chao was a major proponent) and the 拉丁化運動 *lādīnghuà yùndòng* 'Latinization Movement'. In these and other romanization schemes, the issue arose of deciding how to put group the romanized spelling of morphemes into word-sized chunks, something that was of course not faced in the traditional orthography. See (Pan, Yip, and Han, 1993, chapter 5).

The technological include information retrieval (Wu and Tseng, 1993; Palmer and Burger, 1997), text-to-speech synthesis (Sproat et al., 1996; Shih and Sproat, 1996a) and speech recognition. In text-to-speech synthesis (TTS), for example, correct placement of word boundaries is important for assigning appropriate prosody. Such applications require the ability to *automatically* predict word boundaries from an unsegmented stream of characters; as we shall see in later sections, this is not a trivial task.

If one is developing algorithms for Chinese word segmentation, it is desirable to have a standard against which to compare one's progress: in the absence of an orthographic tradition of writing

word boundaries, and in the absence of clear intuitions in many cases of native readers, how do I know that I'm segmenting a text "correctly"? To address this problem, there have been several recently proposed standards for how to determine what a word is in Chinese.

The standards include:

- Mainland Standard (GB/T 13715–92, 1993).
- ROCLING Standard (Huang et al., 1997).
- University of Pennsylvania/Chinese Treebank (Xia, 1999).

1.3.1 Mainland Standard.

The Mainland standard is characterized by lots of specific rules, not all of them explained or justified. For instance, one rule is:

> Nouns derived in 家 *jiā*, 手 *shǒu*, 性 *xìng*, 員 *yuán*, 子 *zi*, 化 *huà*, 長 *zhǎng*, 頭 *tóu*, 者 *zhe* are segmented as one word: 科學家 *kēxuéjiā* 'scientist'

This seems reasonable enough, but for some reason 們 *men* is segmented separately, except in 人們 *rénmen* 'people', and words with *erhua*: 哥儿們 *gērmen* 'brothers'. Personal names are also split: 毛 澤東 'Mao Zedong'.

AABB reduplicants are one word: 高高興興 *gāogāoxìngxìng* 'happy'. On the other hand ABAB reduplicants are two words: 雪白 雪白*xuěbáixuěbái* 'snow white'.

1.3.2 ROCLING Standard.

The ROCLING standard seeks a standard that is "linguistically felicitous, computationally feasible, and must ensure data uniformity." As such, general guidelines are set out from which more specific rules of segmentation can be derived. The two main guidelines are as follows:

1. A string whose meaning cannot be derived by the sum of its components should be treated as a segmentation unit.
2. A string whose structural composition is not determined by the grammatical requirements of its components,or a string which has a grammatical category other than the one predicted by its structural composition should be treated as a segmentation unit.

More specific guidelines include:

1. Bound morphemes should be attached to neighboring words to form a segmentation unit when possible.

2. A string of characters that has a high frequency in the language or high cooccurrence frequency among the components should be treated asa segmentation unit when possible.

3. Strings separated by overt segmentation markers should be segmented.

4. Strings with complex internal structures should be segmented when possible.

Finally, a reference lexicon, based on a reference corpus is assumed.

1.3.3 University of Pennsylvania Standard.

The University of Pennsylvania standard is more akin in spirit to the Mainland standard in that it has less philosophy than the ROCLING standard, and more detailed rules.

One property that is relevant to applications such as TTS is sensitivity to phonological considerations. Thus, while 室內 *shìnèi* 'in the room' is considered to be a single word, 中午 以後 *zhōngwǔ yǐhòu* 'after noon' is considered to be two words, which accords with the sense that 內 *nèi* 'in' functions as an affix whereas 以後 *yǐhòu* is phonologically a separate word.

Another property that sets the UPenn standard apart from the others is that internal structure is marked. So 打得破 *dǎ-dé-pò* 'able to break' is tagged as [打/V 得/DER 破/V]/V, or in other words, as a complex word with internal structure.

1.3.4 Comparison between the systems

A detailed comparison between the three standards is beyond the scope of this discussion, but it is useful to consider a few differences, which we lay out in Table 2, adapted from (Xia, 1999):

Form	UPenn	Mainland	ROCLING	Example
ABAB	ABAB	AB-AB	ABAB	研究研究 'research (a bit)'
AA-看	[AA/V kàn/V]/V	AA kàn	AA kàn	說說看 'talk about it and see'
Pers. Names	One Seg	Two Segs	One Seg	史立璿 Shi Lixuan
Noun + 們	One Seg	Two Segs	Two Segs	朋友們 'friends'
Ordinals	One Seg	Two Segs	Two Segs	第一 'first'

Table 2: Some differences between the segmentation standards, from (Xia, 1999).

To date, only relatively small standard corpora have been segmented according to the criteria proposed by these standards. For instance, at the time of writing, the UPenn standard has been applied to only 100K words; the ROCLING standard has been applied to somewhat more — 5 million words.

1.4 Preview: What Corpus-Based Methods Can Do For Us

So far we have only talked descriptively about morphological processes and their productivity. In the remaining sections on morphology we will consider what corpora of texts can tell us about the morphology of Chinese.

One thing such corpora can give us a clue to is how strongly a sequence of characters coheres as a unit, or in other words how strongly a sequence of characters are associated with one another. While such information does not automatically tell us that the sequence is a word, it at least gives us a clue that perhaps the sequence in question is a lexical item. We will start in the next section with a discussion of measures of association.

We can also examine the productivity of morphological processes. In various places in this first section we have mentioned that such and such a process is productive. We will show that intuitions about productivity often have a statistical interpretation that can be derived from the distribution of words of a particular morphological class in corpora.

While it does not relate to morphology per se, it will also be worthwhile taking a brief look at word frequency distributions.

Corpus-based methods have also been argued to give us a handle on some otherwise hard to pin down processes such as *suoxie*.

Finally, statistical corpus-based approaches have a practical application to various natural-language processing tasks. In the case of Chinese, one important such application is to word segmentation.

2 Statistical Measures

In this section we cover three applications of statistical methods. The first is the distributions of word frequencies, the second morphological productivity, and the third measures of association between terms.

The discussion in this section presumes some knowledge of elementary statistics. A good introduction to the statistics relevant for the statistical analysis of language can be found in the introductory chapters of (Manning and Schütze, 1999).

2.1 Properties of Word Frequency Distributions

As is well-known, a histogram of the words in any reasonable-sized corpus for any language will reveal a distribution that is often described as *Zipfian*, named after George Kingsley Zipf, one of the earliest students of word frequency distributions (e.g. (Zipf, 1949)). Zipf's Law states that the frequency of a word $f(w)$ is proportional to the inverse of its rank $\frac{1}{r}$, where the rank is simply the position in which the word occurs in an inverse frequency ordered list:

(48) $f(w) \propto \frac{1}{r}$

Figure 1 shows a typical word frequency distribution, plotted on a log-log scale, for the 37 million words of the 1995 Associated Press newswire. Note that the log-log plot in Figure 1 is almost — but not completely — straight. At the top of the plot, the distribution starts off almost flat; this part of the distribution is accounted for by just the first ten or so highest frequency words. At the bottom, we see quantization at the lowest frequencies (highlighted by the fact that this is a log-log plot). The lower portion of the curve also shows a slight bulge. Still, this curve is relatively flat, and to see this compare Figure 1 with Figure 2, which plots the distribution for Chinese *characters* for the 10 Million character ROCLING corpus. In this case the curve is markedly bulged, indicating that most characters occur a lot (100 or more times), and only a small number occur infrequently: this is characteristic of a situation where the number of basic elements is fixed, and most of the elements are "saturated".

The Zipfian character of typical word distributions leads to a situation where one has lots of words that occur just a few times, or in other words where there is a *Large Number of Rare Events*. For the 1995 Associated Press corpus, for instance, fully 40% of the word types occur just once, a fairly typical proportion. (In contrast among the 10 Million character ROCLING corpus, only 11% of the *characters* occur once.) For a smaller corpus, such as the Brown corpus (1 Million words), the amount will be closer to 50%.[3] The fact that samples of language contain a large number of rare events has important implications for the statistical treatment of text. For example,

[3]Note that this may in part be due to the design of the Brown corpus, which consisted of a large number of small selections from a variety of genres: one may therefore be seeing the effects of an oversampling of words that are relatively genre-specific, and thus occur in one sample, but nowhere else in the corpus.

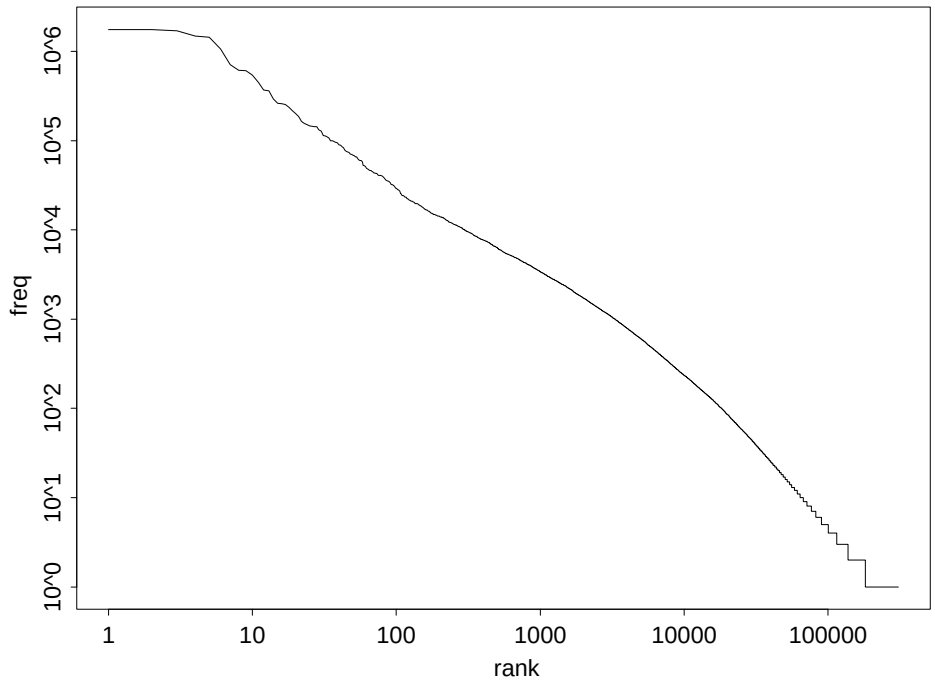


Figure 1: Rank (x) plotted against frequency (y) on a log-log scale for the 1995 Associated Press.

Dunning (1993) argues that many measures of association that have been proposed, such as mutual information (see Section 2.3.2) are inappropriate since they assume a normal distribution of words, which is not the case.

Another important property of word frequency distributions is that the parameters change with the corpus size. Clearly for increasing corpus size N (i.e., the number of word *tokens* in one's sample), the size of the vocabulary (the number of word *types*) $V(N)$ increases. This accords with intuition: the more text one looks at, the more words one expects to see; see Figure 3, Panel A. But as Baayen (2001) notes, other parameters also increase with increasing sample size. Thus one can measure the mean frequency of words as $\frac{N}{V(N)}$. This, too, increases with sample size; see Figure 3, Panel B. This also makes intuitive sense: as N continues to increase, you will continue to see new words, but the larger N gets, the fewer new words one expects to see in the next unseen sample. So $V(N)$ grows, but not as quickly as N. Thus $\frac{N}{V(N)}$ increases with increasing N. Much of Baayen's discussion is devoted to what he terms the "quest for characteristic constants", that is measurable properties of corpora that are invariant with corpus size. As he discusses, there have been various proposals of varying degrees of mathematical sophistication, but on the whole the quest has not been successful. The moral of this story is that any parameters that one collects from a corpus are likely to be affected by the sample size.

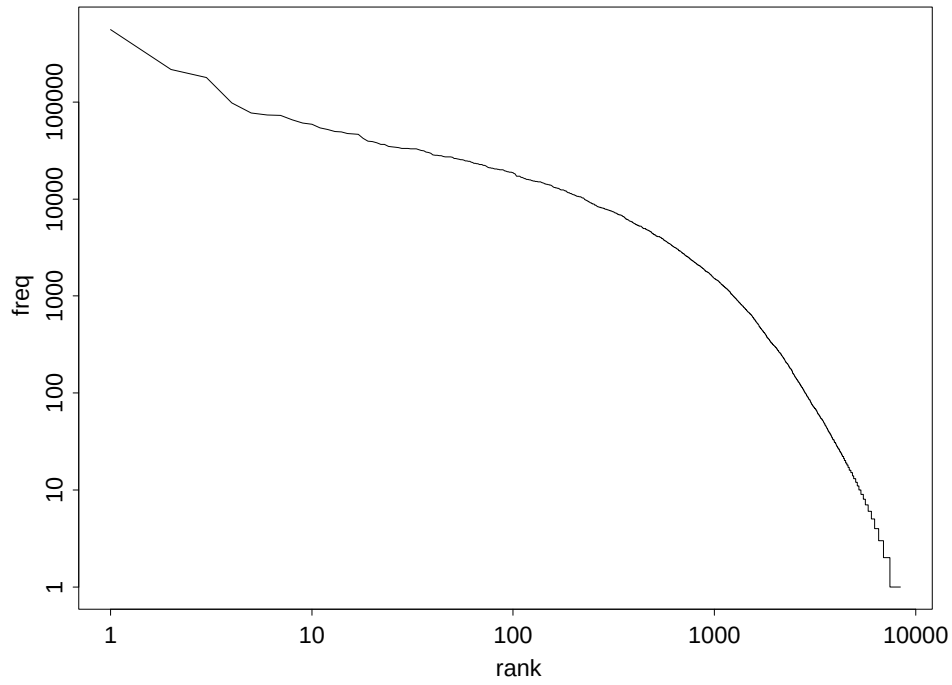


Figure 2: Rank (x) plotted against frequency (y) on a log-log scale over *characters* for the 10 Million character ROCLING corpus.

2.2 Measures of Morphological Productivity and a Case Study

One of the most interesting results of work on corpus-based studies of morphology over the past decade or so has been the development of robust measures of morphological productivity; much of this work has been done by Baayen and his colleagues.

Various measures of morphological productivity have been proposed in the literature. One measure, due to Aronoff (1976), and discussed in (Baayen, 1989) is defined as:

(49) $I = \frac{V}{S}$

where V is the number of distinct instances of a morphological category — e.g., all words in an English dictionary ending in the suffix *-ness* — and S is the number of *potential* types of that category. The idea, clearly, is that if an affix, say, is productive, one will find that a lot of the formations that the affix might potentially form will in fact be found.

The problem of course is to measure V and S. V cannot be measured simply by counting the words of a given morphological category found in a dictionary: as Baayen observes, dictionaries are typically unreliable indicators of what words there are in the language, particularly when it

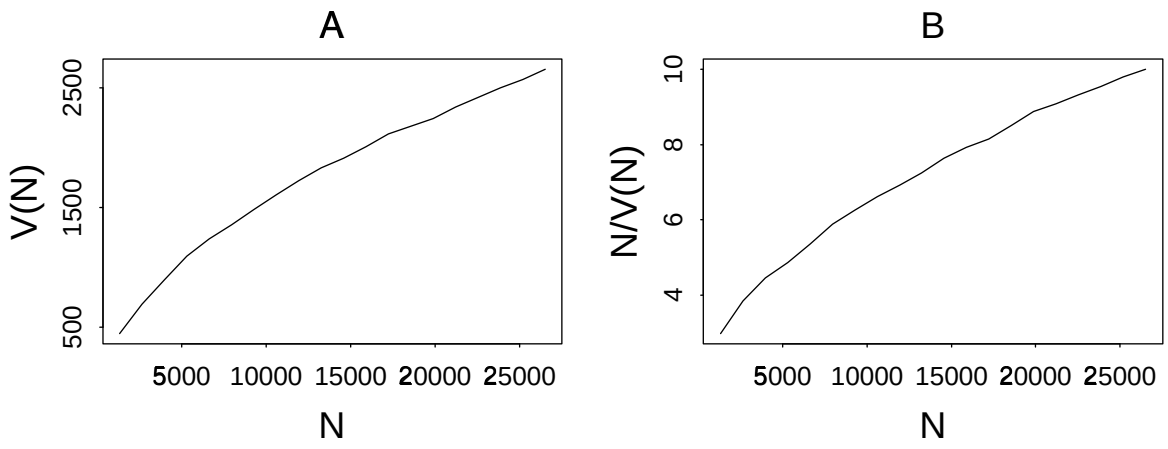


Figure 3: Vocabulary size $V(N)$ (Panel A) and mean word frequency $\frac{N}{V(N)}$ (Panel B) as a function of sample size N in *Alice in Wonderland*, measured at 20 equally spaced intervals. From (Baayen, 2001), figure and caption kindly provided by Harald Baayen.

comes to productive formations: it is not worth the space or time to develop long lists of entries for words where the formation is predictable and therefore presumably known to the (native) reader anyway. One might of course measure V by counting the types of a given category found in a corpus, but still one is likely to miss valid instances that simply fail to occur in the corpus. It is even more unclear how to estimate S: the English suffix *-ness*, would seem to be able to attach to any adjective, so one might reasonably calculate S by first computing the number of adjectives (ignoring for the sake of argument the fact that this itself is hard to estimate). But what about a (seemingly less productive) deadjectival-noun forming affix *-ity*. Is S to be computed under the assumption that *-ity* attaches to any adjective? Perhaps we should take into account the fact that *-ity* seems to attach almost exclusively to latinate adjectives and lower our estimate of S. But then what about *-th*, as in *width*, or *breadth*? This is evidently a very unproductive affix in English, but on what basis do we draw that conclusion: perhaps *-th* is legitimately restricted to the set of native monosyllabic adjectives that denote physical properties, a very small set. In that case the measure $\frac{V}{S}$ might actually be somewhat large.

A measure that turns out to be reasonable in that it has a well-defined method of computing, and accords with intuition as well as more rigorous psychological measures of productivity is that given in (50) (Baayen, 1989; Baayen and Lieber, 1991). Here N is the number of *tokens* of a particular construction found in a particular corpus: for example, the number of tokens of all nouns ending in *-ation*. The term n_1 represents the number of hapax legomena, the types of that construction that

are represented in the corpus by a single token.

(50) $\mathcal{P} = \frac{n_1}{N}$

This measure is the same as what Church and Gale (1991) call the 'Good-Turing' Measure, and originates in work of Good (1953) on population biology. $\mathcal{P}$ is actually an estimate of the probability that the next token that we see of a particular construction will be one that we have never seen before.

One can get an intuitive feel for why this ratio gives a good approximation of this probability by considering the standard probability-textbook metaphor of an urn full of balls. Consider the bucket in Figure 4 containing a large number of balls, and suppose we have sampled these balls as shown so that in our sample we have 9 green balls, 8 red balls, 6 orange balls and four blue balls. If our sample could be argued to be large enough, it would be reasonable for us to suppose that if we were then to continue drawing balls we would find that there were only four colors, and that they would continue to be found in the ratio **9**(green): **8**(red): **6**(orange): **4**(blue).

Figure 4: A collection of balls where all ball types have been seen.

Now suppose that we have a different bucket, as shown in Figure 5, and we do the same experiment, but this time we find that in addition to a number each of green, red, orange and blue balls, there are colors which are only represented by two balls in the sample, and even more colors that are only represented by one ball in the sample. In this case, again assuming our sample is sufficiently large, we would be much less confident that we had seen all the colors that we are going to see. Indeed, the more singleton colors we find, the more likely we would be to assume that if we continue to draw balls, we will continue to find new colors. Now obviously we would want to scale our estimate of the probability of finding new colors by the size of the sample we have chosen: if we find 20 singletons out of a sample of 100 balls, we would expect a higher probability of seeing new colors than if we saw 20 singletons out of a sample of 1000 balls. But this scaled estimate clearly involves the ratio n_1/N.

Figure 5: A collection of balls where many ball types have not been seen.

To complete the story we need to make the connection between the probability estimate n_1/N, and the notion of productivity of a particular morphological category. But this is elementary since the more productive a category is, the higher our expectation that the next instance of that category we see will be novel. Thus $\mathcal{P}$, as defined above seems to be a reasonably intuitive measure of morphological productivity. See (Baayen, 1989; Baayen, 1993) for more extensive justification of n_1/N as one measure of productivity, and for some comparison with other conceivable measures.

To give a sense of the kinds of values of $\mathcal{P}$ that are associated with productive morphemes, we list in (51) some examples of productive Dutch morphological processes from (Baayen, 1989):

(51)

Morpheme	Gloss	$\mathcal{P}$
-ing	'-ion, -ing'	0.038
-heid	'-ity'	0.114
noun compounds	—	0.225

Comparable values from (Baayen and Lieber, 1991) for some English morphological processes are given below. Simplex nouns are included since it is useful to compare the productivity of morphological processes as estimated by $\mathcal{P}$, with the estimated productivity of what is putatively a closed class:

(52)

Morpheme	$\mathcal{P}$
-ness	0.0044
-ish	0.0034
-ation	0.0006
-ity	0.0007
simplex nouns	0.0001

Returning to Mandarin, note that $\mathcal{P}$ for the somewhat productive (though restricted) noun plural affix 們 *-men* is about 0.04; for the corpus over which this value was computed, $N = 5948$ and

$n_1 = 247$. Similarly, $\mathcal{P}$ for the very productive experiential verbal affix 過 *-guò* is 0.20, based on a sample where $N = 1666$ and $n_1 = 330$. (The reader is invited to compare these $\mathcal{P}$ values with the $\mathcal{P}$ values for the more productive nominal roots given below in Table 4.) One can compare these $\mathcal{P}$ values with that of single-character nouns — an obviously closed and non-productive class — which is effectively 0.

Lab Exercise 1 for this section deals with measuring the productivity of various Mandarin affixes on the Penn Chinese Treebank corpus.

2.2.1 Mandarin Root Compounds: A Case Study in Morphological Productivity

(The following discussion is a summary of the salient points of (Sproat and Shih, 1996).)

Following Anderson's (1992) analysis of compounds in English, Dai (1992) argues that compounding in Mandarin always consists of the composition of free words, never bound forms.[4] Thus while the example in (53) is a compound, the example in (54) is not:

(53) 哈密瓜 *hāmìguā* (Hami melon) 'cantaloupe'

(54) 香腸 *xiāngcháng* (fragrant intestine) 'sausage'

This is because in 哈密瓜 *hāmìguā*, 'cantaloupe', both 哈密 *hāmì* 'Hami' and 瓜 *guā* 'melon' are words. But in 香腸 *xiāngcháng* 'sausage', 香 *xiāng* 'fragrant' is a word, but 腸 *cháng* 'intestine' is not a word, but rather a bound root: the normal Mandarin word for 'intestine' is 腸子 *chángzi*.

The problem is that such bound roots seem to play an active role in the morphology of Mandarin, occurring in a large number of formations. A few more examples follow, where in each case the full form of the word is given first with the nominal root underlined, and then a derived word is illustrated:

(55) 螞蟻 *mǎyǐ* 'ant'
工蟻 *gōngyǐ* 'worker ant'

(56) 腦子 *nǎozi* 'brain'
腦水腫 *nǎoshuǐzhǒng* (brain water swelling) 'hydrocephaly'

(57) 蒼蠅 *cāngyíng* 'fly'
地中海蠅 *dìzhōnghǎiyíng* 'Mediterranean fly'

(58) 蘑菇 *mógū* 'mushroom'
菇傘 *gūsǎn* (mushroom umbrella) 'pileus'

[4] Note that Dai's notion of compound accords with what (Packard, 2000) calls compound, though Packard, following (Sproat and Shih, 1996), does acknowledge the existence of productive non-word-based morphological formations.

Not only do the formations seem productive, but they are not constrained to one position or another, belying any notion that these might be some form of affix, since affixes typically attach in a predetermined position. See (59), repeated from (25):

(59)

蟻 *yǐ*	蟻王	工蟻
	yǐwáng ‘queen ant’	*gōngyǐ* ‘worker ant’
腦 *nǎo*	腦水腫	後腦
	nǎoshuǐzhǒng ‘hydrocephaly’	*hòunǎo* ‘hindbrain’
蠅 *yíng*	蠅屍	地中海蠅
	yíngshī ‘fly corpse’	*dìzhōnghǎiyíng* ‘Mediterranean fly’
菇 *gū*	菇傘	金菇
	gūsǎn ‘pileus’	*jīngū* ‘golden mushroom’

This appears to suggest that such constructions *must* be compounds: theoretical prejudices aside, what else could they be? However, if it turned out that they were not productive, one might perhaps discount them. So are they productive?

Table 3 shows all examples of derived compounds involving the root 菇 *gū* ‘mushroom’ collected from a 40 million character corpus.[5] While some forms, such as 香菇 *xiānggū* ‘mushroom’ and 蘑菇 *mógū* ‘mushroom’ occur very frequently, 19 (38%) of the types occur just once.

Results are presented in Table 4. For each root we present the canonical whole word derived from that root, along with the values for:

- n_1
- N
- V: the number of *types* in our sample, corresponding to Baayen’s (1989) notion of *pragmatic usefulness*
- P_{max}: the token count for the most frequent type, and
- $\mathcal{P}$

The data represent all (relevant) examples found in the database with the exception of the noun roots 石 *shí* ‘rock’, and 腦 *nǎo* ‘brain’, which exhibit too many derivatives in the database for hand-checking to be practical: for these roots we took random subsamples. As the data in Table 4 show, roots ranging from 石 *shí* ‘rock’ to 肚 *dù* ‘belly’ form root compounds with some degree of productivity.

It is worth noting that in some cases the value for $\mathcal{P}$ was significantly diminished by the presence of a single type with extremely high token frequency. The most dramatic case is with the root 腦 *nǎo* ‘brain’, where the P_{max} value of 791 is associated with the word 電腦 *diànnǎo* ‘computer’.

[5] Provided by United Informatics, Inc.

224	香菇	104	蘑菇	102	洋菇
23	菇類	16	菇農	9	磨菇
9	草菇	8	菇寮	8	冬菇
8	塊菇	7	金針菇	6	出菇
5	菇價	4	菇傘	4	菇柄
4	鮑魚菇	4	慈菇	3	夏塊菇
3	裸蓋菇	3	茨菇	3	冬季菇
2	菇舍	2	菇木	2	鮮菇
2	發菇	2	採菇	2	夏季菇
2	食用菇	2	食菇	2	春季菇
2	出菇期	2	三菇菜膽	1	菇體
1	菇醇	1	菇腳	1	菇菌
1	菇湯	1	菇場	1	鮑菇
1	褐菇	1	種菇	1	黑菇
1	菌菇	1	產菇量	1	乾菇
1	金菇	1	早發菇	1	生菇
1	可食菇	1	包菇	1	山菇

Table 3: Distribution for the Mandarin nominal root *gū* 'mushroom' collected from a 40 million character corpus.

In other words, 73% of the tokens derived from 腦 *nǎo* 'brain' can be attributed to this one type. One is almost justified in removing 電腦 *diànnǎo* 'computer' from our data, both because it is a statistical outlier, and because it clearly does not belong to the 'brain' class on semantic grounds. Were we to do this, the $\mathcal{P}$ value for 腦 *nǎo* 'brain' would increase dramatically to 0.12. In general, for those roots where P_{max} represents a large percentage of N, the productivity is almost certainly being underestimated by the $\mathcal{P}$ values we have presented.

So at least some roots seem to be fairly productive, somewhere between the productivity of a very productive inflectional affix such as the experiential affix 過 *-guò* and the less productive nominal plural affix 們 *-men*. This suggests that root compounding is in principle productive, contrary to Dai's (1992) claim.

The only caveat in comparing the numbers in these data with productivity measures for other data is that it is difficult to make comparisons between datasets of different sizes (Baayen, 2001).

2.3 Measures of Association

An important area of research in statistical natural language processing over the past decade has been measures of association between words and other linguistic units. And an important application of such measures has been computational lexicography, and in particular the question of

Root	Whole Word	Meaning	n_1	N	V	P_{max}	$\mathcal{P}$
石 *shí*	石頭 *shítou*	'rock'	75	583	147	57	0.129
盒 *hé*	盒子 *hézi*	'box'	82	1205	167	257	0.068
蟻 *yǐ*	螞蟻 *mǎyǐ*	'ant'	21	322	35	173	0.065
蛙 *wā*	青蛙 *qīngwā*	'frog'	22	407	49	199	0.054
龜 *guī*	烏龜 *wūguī*	'turtle'	21	414	46	137	0.051
餃 *jiǎo*	餃子 *jiǎozi*	'dumpling'	8	167	19	66	0.048
蠅 *yíng*	蒼蠅 *cāngyíng*	'fly'	14	325	35	142	0.043
棉 *mián*	棉花 *miánhuā*	'cotton'	52	1283	138	147	0.041
菇 *gū*	蘑菇 *mōgū*	'mushroom'	19	598	49	224	0.032
木 *mù*	木頭 *mùtou*	'wood'	265	8904	617	701	0.030
腦 *nǎo*	腦子 *nǎozi*	'brain'	34	1077	75	791	0.032
駝 *tuó*	駱駝 *luòtuó*	'camel'	6	210	16	104	0.029
腸 *cháng*	腸子 *chángzi*	'intestine'	62	2268	148	373	0.027
蜂 *fēng*	蜜蜂 *mìfēng*	'bee'	23	858	63	104	0.027
肚 *dù*	肚子 *dùzi*	'belly'	36	1434	83	734	0.025

Table 4: Productivity measures of some Mandarin nominal roots, measured over a 40 million character corpus.

what "collocations" are sufficiently interesting to merit possible inclusion in a lexicon, whether for human consumption or for other NLP applications. Early work such as (Church and Gale, 1991) was advertised as being, in effect, a lexicographer's helper, a semiautomatic technique for deciding which collocations of English words might be of interest to a lexicographer.

In the Chinese context, measures of associations are useful not only for discovering phrasal collocations, but also for helping to decide on what groups of characters might be considered words. Indeed, one of the earliest applications of such collocational methods to Chinese, (Sproat and Shih, 1990), was a direct application of Church and Gale's work to the problem of automatically detecting two-character words.

In this section we discuss various measures of association, considering the merits and limitations of each. Most of these association measures are designed for determining the association between two terms. However, at the end of this section we will discuss some methods that enable one to detect multi-term collocations.

In the ensuing discussion, we will restrict ourselves to examples from English: computing the various measures of association for Chinese will be part of the laboratory exercise for this section.

The presentation of association measures presented here is based on the presentation in (Manning and Schütze, 1999); useful discussion of relevant issues can also be found in (Weeber, Vos, and Baayen, 2000).

	AP92		AP93		AP94	
	f	p	f	p	f	p
the	1,659,949	0.039	1,451,984	0.039	1,311,237	0.039
United	30,883	0.00072	28,336	0.00076	25,456	0.00075
country	21,225	0.00050	18,304	0.00050	15,919	0.00047
night	12,422	0.00029	10,328	0.00028	10,566	0.00031
dog	1,048	2.44e-05	1,045	2.80e-05	992	2.91e-05

Table 5: Maximum likelihood estimates for five common words from three years of the Associated Press (1992–1994). N is 43,012,596, 37,386,960, and 34,041,151, respectively, for these three years.

2.3.1 Probability Estimates

Measures of association between terms invariably depend upon estimates of the probability of the individual terms, as well as the probability of the terms cooccuring in some context. Probabilities are estimated from corpora, and the simplest estimate of $p(t)$, the probability of a term t is is the *maximum likelihood estimate* $f(t)/N$, where $f(t)$ is the frequency of t in the corpus, and N is the corpus size. This estimate is pretty good for frequent words. For example if you look at the probability estimates for a set of frequent words in three consecutive years of the Associated Press newswire, we find that the probability estimates are pretty similar. See Table 5.

(Clearly in such estimates one wishes to avoid names, and other words that are likely to be topical to a particular corpus. If one looked at the Associate Press newswire from 1989 through 2001, one would find that *Bush* was very frequent from 1989 until 1993, at which point *Clinton* would become very frequent, until 2001, when *Bush* would become frequent again.)

The problem with the maximum likelihood estimate is that it becomes very bad for small counts: the true probability of a word that occurs once in a corpus is almost certainly not well estimated by $1/N$. Even worse are events that never occur in the corpus: the maximum likelihood estimate would give the probability of these events as 0, whereas the probability is likely not to be 0.

A huge amount of research has gone into various ways of providing better estimates for low probability events, and these typically involve various ways of *smoothing* the probability distribution initially estimated by the maximum likelihood estimate. Manning and Schütze (1999) describe a number of these in Chapter 6 of their book. One of the best they describe is the Good-Turing estimator, which we have already seen in the guise of a measure of productivity. For an observation (i.e., either a token or an ngram) with frequency r, we provide a reestimate r^* defined as follows:

$$(60) \quad r^* = (r+1)\frac{E(n_{r+1})}{E(n_r)}$$

where $E(n_r)$, is the expected number of tokens with frequency r. The expected probability mass of the *unseen* items (items where $r = 0$) is given as $E(n_1)/N$; this is just the measure of productivity

that we saw earlier.

There are a couple of ways of using Good-Turing. One is to actually estimate $E(n_{r+1})$ using a smoothing function. Another and more direct way is to use the actual observations of n_r for $E(n_r)$. For low-frequency words, this estimate will be reasonably accurate: the number of words seen, say, twice in any corpus will be large so n_2 will be a reasonable estimate of the true expectation of n_2. For low frequency words we can thus re-estimate their frequency as:

$$(61) \quad r^* = (r+1)\frac{n_{r+1}}{n_r}$$

For frequencies other than zero this will result in a reestimation downwards. The probabilities will then be reestimated as $\frac{r^*}{N}$, with the probability mass $\frac{n_1}{N}$ reserved for unseen items.

For high frequency items estimating $E(n_r)$ as just n_r is typically bad, since the number of events with a given frequency is small — often 1 — for high frequency events. However, for high frequency words the maximum likelihood estimate is a reasonable estimate of their probability anyway, so a reestimation is not really needed. (Note that one does need to make sure, under any reestimation, that the resulting probabilities all add up to 1: this can generally be handled by renormalizing, dividing each probability by the sum of the unrenormalized probabilities.)

2.3.2 (Specific) Mutual Information

Mutual Information was originally proposed as an information-theoretic measure of channel capacity (Fano, 1961). It was introduced into computational linguistics by Church and Gale (1991), as a method for finding lexicographically interesting collocations, and has since become arguably the most popular measure of association. The basic idea behind mutual information is that two events that are highly associated should show a probability of occurring together that is high relative to the probability of them occurring together by chance. Given two events x and y with associated probabilities $p(x)$ and $p(y)$, the expected probability of the two events cooccurring is, given classical probability theory, simply $p(x)p(y)$. If the actual measured probability, denoted $p(x, y)$ is high compared to $p(x)p(y)$, then we would say that the events are more highly associated with each other than would be expected by chance. The mutual information of x and y — $I(x; y)$ — defined in (62), is just the base 2 log of this ratio:

$$(62) \quad I(x; y) = \log_2 \frac{p(x, y)}{p(x)p(y)}$$

As an example of the kind of word associations derived via mutual information, consider Table 6, which presents a sample of adjacent words that have a relatively high mutual information, derived from the 1995 Associated Press (37 million words). Here probability is computed using the maximum likelihood estimate; in this case we restrict ourselves to words that occur more than a hundred times, so the maximum likelihood estimate is pretty reasonable. It will be seen that many of these pairs (though surely not all) are reasonable collocations, either proper names (or parts of proper names) or else phrases one might expect to find in a dictionary.

M.I.	f(1,2)	f(1)	f(2)	pair
18.3908	101	104	106	Picket Fences
18.3280	101	101	114	Highly Effective
18.2061	119	121	122	Ku Klux
18.1722	124	124	127	alma mater
18.1574	115	124	119	SPACE CENTER
18.1480	118	127	120	JUDGE LANCE
18.1275	127	131	127	Phnom Penh
18.1266	124	126	129	Velika Kladusa
18.0901	95	103	124	Ginny Terzano
18.0627	134	136	135	Notorious B.I.G
18.0580	116	119	134	Spiritual Laws
18.0486	123	134	127	Deepak Chopra
18.0271	80	105	107	Myriam Sochacki
17.9417	147	149	147	TEL AVIV
17.8421	96	117	131	Reba McEntire
17.7610	108	152	120	Dollar Spin
17.7551	117	124	160	SALT LAKE

Table 6: Sample of highly associated adjacent word pairs from the 37 million words of the 1995 Associated Press. Shown are, from left to right: the mutual information (M.I.); the frequency of the pair f(1,2); the frequency of the first word f(1) and the frequency of the second word f(2). Note that f(1) and f(2) are both greater than 100 for this sample.

In contrast, Table 7 shows some examples of poorly associated words, also from the 1995 AP. In these cases, the terms seem to have little particularly to do with each other, and one would not expect to find the pairs in a dictionary.

As we will see in a later section, mutual information has been proposed as, among other things, a method for automatically segmenting Chinese text (Sproat and Shih, 1990).

Nonetheless, despite its popularity, mutual information has a couple of problems, as Manning and Schütze (Manning and Schütze, 1999) note. First of all it is unreliable for small counts: if two terms t_1 and t_2 cooccur with frequency 2, and if their respective frequencies are also 2, their mutual information will be quite high (to be exact, it will be $\log_2(N) - 1$, assuming the maximum likelihood estimate for probability). But really the evidence that the terms is associated is weak: this is not a problem with mutual information per se, but rather with the maximum likelihood estimate, since for small counts, any measure that is based on an estimate of probability is likely to be suspect.

The second, and more serious problem is that mutual information depends upon estimated probability (or frequency) in a counterintuitive way. Suppose that two words w_1 and w_2 occur each

M.I.	f(1,2)	f(1)	f(2)	pair
-0.000104371	2	44293	1694	But 32
-0.000104371	12	44293	10164	But public
-0.000108717	1	5938	6318	big reports
-0.000122066	7	540363	486	in heroin
-0.000122066	7	486	540363	determination in
-0.000123791	1	20716	1811	says Hall
-0.000135193	2	567	132335	Building from
-0.000135827	1	6639	5651	water given
-0.000135827	1	5651	6639	given water
-0.000137212	1	8365	4485	programs enough
-0.000137212	1	2275	16491	village children
-0.000144883	2	5283	14203	study most
-0.000151325	1	14452	2596	her includes
-0.000163745	1	645	58167	Delaware this

Table 7: Sample of poorly associated adjacent word pairs from the 1995 Associated Press.

100 times, and that they also cooccur 100 times: that is, one word only occurs when the other occurs, and they are thus "perfectly" associated. If the corpus size is 10 million words, then the mutual information of the two words is given by:

$$I(w_1; w_2) = \log_2\left(\frac{\frac{100}{10{,}000{,}000}}{\frac{100}{10{,}000{,}000}\frac{100}{10{,}000{,}000}}\right)$$

or: $I(w_1; w_2) = \log_2(100{,}000) = 16.6$

Now let's suppose that w_1 and w_2 each occur 1,000 times, and again only with each other. If this occurs in the same size corpus, we would surely think that w_1 and w_2 are even more highly associated than if they had only occurred 100 times. Unfortunately, the mutual information in this instance would be lower, not higher, which is, as Manning and Schütze note, exactly the opposite of what we would expect from an association measure:

$$I(w_1; w_2) = \log_2\left(\frac{\frac{1{,}000}{10{,}000{,}000}}{\frac{1{,}000}{10{,}000{,}000}\frac{1{,}000}{10{,}000{,}000}}\right) = \log_2(10{,}000) = 13.3$$

2.3.3 Frequency-Weighted Mutual Information

One solution to the counterintuitive aspect of mutual information mentioned above is to use *frequency-weighted mutual information*, defined as follows:

F.W.M.I.	f(1,2)	f(1)	f(2)	pair
469631	174742	708948	1435262	of the
341288	129132	540363	1435262	in the
184196	16797	16816	18733	Associated Press
182881	17287	24135	17564	United States
181021	66059	258389	1435262	to the
144801	31575	447529	110206	to be
140778	51336	200524	1435262	on the
136748	12956	24177	13364	New York
135747	19968	112171	60003	have been
135610	12452	17858	13778	All Rights
134177	28748	144059	294607	he said
133324	11684	13778	11684	Rights Reserved
120476	17592	95448	60003	has been
118810	50201	254405	1435262	for the
114737	17820	69930	110206	will be
109856	15576	261695	16816	The Associated
107843	36922	127433	1435262	at the
97455	9561	10928	28038	White House
96982	15799	76338	110206	would be

Table 8: Sample of highly associated adjacent word pairs from the 1995 Associated Press. Shown are, from left to right: the frequency weighted mutual information (F.W.M.I.); the frequency of the pair f(1,2); the frequency of the first word f(1) and the frequency of the second word f(2). Again, f(1) and f(2) are both greater than 100 for this sample.

(63) $$I_{fw}(x;y) = f(x,y)\log_2 \frac{P(x,y)}{P(x)P(y)}$$

Thus, we simply multiply the standard mutual information measure by the frequency of the pair. In this way, any loss of mutual information due to an increase in the frequency of both the pair and the unigrams, is offset by the factor for the pair.

Table 8 gives examples of collocations found in the 1995 Associated Press using frequency weighted mutual information.

The problem with frequency-weighted mutual information, as will be readily observed in Table 8, is that it tends to over-reward frequency. While some obviously frequent collocations — *Associated Press*, *United States*, *New York*, *White House* clearly make sense as things to list in a lexicon — many of the collocations found by the method are merely common syntactic combinations. No one would presumably consider *of the*, *in the*, *to the*, *on the*, *have been* or *he said* to be

	$w_1 =$ *new*	$w_1 \neq$ *new*
$w_1 =$ *companies*	8 (*new companies*)	4,667 (e.g. *old companies*)
$w_1 \neq$ *companies*	15,820 (*new machines*)	14,287,181 (e.g. *old machines*)

Table 9: A 2×2 table showing the distribution of bigrams in a corpus (from (Manning and Schütze, 1999, Table 5.8, page 169)). There were 8 instances of *new companies*, 4,667 instances of *X companies*, where *X* is different from *new*, 15,820 instances of *new Y*, where *Y* is different from *companies*, and 14,287,181 instances of *XY*, where *X* and *Y* are different, respectively, from *new* and *companies*.

interesting units from a lexicographic point of view.[6]

2.3.4 Pearson's χ-Square

The chi-square test is a statistical test, like the t-test, that provides a confidence measure for rejecting an assumption of independence between some events. The advantage over the t-test is that the chi-square test seems to be more robust than the t-test for distributions that violate normality; see (Manning and Schütze, 1999, page 169).

The chi-square test is applied to tables; for example the simple 2×2 table in Table 9, copied from Manning and Schütze's discussion (page 169), showing the distribution of bigrams *new companies*, *new Y*, *X companies* and *X Y*, where *X* and *Y* are different, respectively, from *new* and *companies*.

The general formula for chi-square is given by:

$$\chi^2 = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{64}$$

where O_{ij} is the observed value in table cell ij, and E_{ij} is the expected value.

The expected value of a cell is computed, under the assumption of independence, as the product of the marginal probabilities of the row and the column containing that cell. In the case of *new companies*, where the total number of bigrams is $N = 8 + 4667 + 15820 + 14287181 = 14307668$ we can estimate this by the probability of *new* ($\frac{8+15820}{14307668}$) and the probability of *companies* ($\frac{8+4667}{14307668}$) times the number of bigrams. The expression for a 2×2 table is (Manning and Schütze, 1999, (5.7), page 170):

$$\chi^2 = \frac{N(O_{11}O_{22} - O_{12}O_{21})^2}{(O_{11} + O_{12})(O_{11} + O_{21})(O_{12} + O_{22})(O_{21} + O_{22})} \tag{65}$$

[6] Still it has to be noted that in many languages such constructions do indeed form single words. So for example in Irish *in the* which one would expect to be *í an* (in the) is actually expressed by a single suppletive form *sa*. The same is true in German constructions such as *zum* 'to the'.

Some examples of associations computed by chi square from the 1995 Associated Press are give in Table 10.

There are various problems with chi-square.[7] Apart from assumptions of normality of the distributions, which are violated with small counts,[8] there is the problem is that since chi square is a symmetric measure, it does not distinguish between events that are much more likely than chance from events that are much *less* likely than chance. For example, if you look at the 1995 Associate Press, at the phrase *tape-recorded conversations* — a plausible collocation — you find that *tape-recorded* occurs 100 times, *conversations* 639 times, and the collocation 7 times, with a reasonably high chi square value of 28,752.8. Similarly for *sickening reminder* the breakdown is, 17 for *sickening*, 307 for *reminder* and 2 for the pair, with a chi square value of 28,747.7. However a very similar chi square value is obtained for *the of*, where *the* occurs 1,435,262 times, *of* 708,948 times, the pair exactly once (due to a typo in the data), yielding a chi square value of 28,744.5. This is not a problem for genuinely non-occuring bigrams, since one can simply not count these, but for bigrams that occur infrequently, chi square does not distinguish between those that are more likely than they should be from those that are less likely than they should be. Mutual information does not have this problem.

2.3.5 Likelihood ratios

Likelihood ratios (Dunning, 1993; Manning and Schütze, 1999) compute the relative likelihood of two hypotheses concerning two events e_1 and e_2:

- Hypothesis 1: $p(e_2|e_1) = p = p(e_2|\neg e_1)$
- Hypothesis 2: $p(e_2|e_1) = p_1 \neq p_2 = p(e_2|\neg e_1)$

Hypothesis 1 simply says that the probability of e_2 occurring given e_1 is indistinguishable from the probability of e_2 occurring given something other than e_2: i.e., the e_2 is not particularly expected (or unexpected) given e_1. Hypothesis 2 says, in contrast, that there is a difference in expectation, and that e_2 is dependent on e_1.

We can estimate the probabilities p, p_1 and p_2 by the maximum likelihood estimate as follows, where c_1, c_2 and c_{12} are, respectively the frequency of e_1, of e_2, and of e_1 and e_2 cooccurring; and N is the size of the corpus:

$$p = \frac{c_2}{N}, p_1 = \frac{c_{12}}{c_1},$$
$$p_2 = \frac{c_2 - c_{12}}{N - c_1}$$

If we assume a binomial distribution

$$b(k; n, x) = \left(\begin{array}{c} n \\ k \end{array} \right) x^k (1-x)^{(n-k)}$$

[7] We thank Martin Jansche for useful discussion on this point.

[8] See `http://citeseer.nj.nec.com/dunning93accurate.html`.

χ^2	f(1,2)	f(1)	f(2)	pair
999319	600	6262	2156	attorney general
996401	47	280	297	Connie Mack
993054	192	552	2522	20th century
991404	36	299	164	OR MUSIC
991243	306	818	4330	atomic bomb
990132	145	1709	466	loan guarantees
986765	18	113	109	Geographic Explorer
985647	375	2599	2058	Catholic Church
985038	67	1540	111	racial slur
984217	84	1379	195	highly publicized
983361	147	284	2902	plead guilty
982709	23	159	127	Justices Antonin
975478	289	7415	433	Sen Arlen
972147	22	116	161	Celtic Journey
970830	322	2808	1426	illegal immigrants
968693	126	2984	206	flight attendant
968378	261	1571	1679	pleaded innocent
968056	20	124	125	Joey Buttafuoco
967695	37	147	361	silicone implants
967506	145	217	3756	three-judge panel
964801	36	228	221	Senior Citizens
961235	60	439	320	Down syndrome
960219	22	163	116	Global Celtic
960197	60	117	1202	surface-to-air missile
959493	690	2832	6565	stock market
958923	177	1326	924	Steve Forbes
957999	723	1484	13778	Human Rights
957512	42	255	271	Silver Spring
955996	42	635	109	Nine finalists
955707	30	157	225	Wind gusts
953203	29	197	168	seldom votes
952778	289	2368	1388	Chechen fighters

Table 10: Sample of highly associated adjacent word pairs from the 1995 Associated Press, using chi-square. Shown are, from left to right: the chi square value; the frequency of the pair f(1,2); the frequency of the first word f(1) and the frequency of the second word f(2). Again, f(1) and f(2) are both greater than 100 for this sample.

then the likelihoods of the two hypotheses, given the observed counts e_1, e_2 and e_{12}, can be computed as:

$$L(H_1) = b(c_{12}; c_1, p)b(c_2 - c_{12}; N - c_1, p)$$
$$L(H_2) = b(c_{12}; c_1, p_1)b(c_2 - c_{12}; N - c_1, p_2)$$

The *log* likelihood ratio then reduces as follows:

$$\begin{aligned}\log \lambda &= \log \frac{L(H_1)}{L(H_2)} \\ &= \log L(c_{12}, c_1, p) + logL(c_2 - c_{12}, N - c_1, p) \\ &\quad - \log L(c_{12}, c_1, p_1) - \log L(c_2 - c_{12}, N - c_1, p_2)\end{aligned}$$

where:

$$L(k, n, x) = x^k(1 - x)^{n-k}$$

Some examples of associations computed by $-2log\lambda$ (which is asymptotically χ^2 distributed, (Manning and Schütze, 1999, page 174)) from the 1995 Associated Press are give in Table 11.

The likelihood ratio seems to provide good evidence for collocational status, but it shares with chi square the problem that, since it tests the hypothesis of whether two events are distinct, in principle it does not care whether one event is much more likely than the other, or much less likely.

Now Manning and Schütze note (footnote 6, page 172) that the case where p_1 is significantly less than p_2 — i.e., where the probability of w_2 given w_1 is much less than the probability of w_2 under other conditions — is rare. But note that all that is needed for this to occur is that w_1 and w_2 are very frequent, whereas w_1w_2 is very infrequent. In that case $p(w_2|w_1)$ would be low since the probability of seeing w_2 after w_1 would be low, whereas $p(w_2|\neg w_1)$ would be very high (since $p(w_2)$ is by assumption high). So then $p_1 = p(w_2|w_1)$ would be much less than $p(w_2|\neg w_1)$, and Hypothesis 2 would be much more likely, resulting in a high likelihood ratio. To see such an instance, consider the collocations of *powerful* in Table 12, and consider the underlined example *powerful the*. To be fair, this is just one instance out of a collection of otherwise reasonable candidates, but it does suggest that one still needs to be careful in trusting the results of even the best measures of assocation.

2.3.6 Extracting Non-Binary Collocations

The association methods discussed above are designed for computing associations between pairs of terms. It is worth noting that methods have been developed for finding interesting collocations involving more than two words.

There have been extensions of measures such as mutual information to more than two terms. For example (Chang and Su, 1997) define mutual information for three terms as:

$-2\log\lambda$	f(1,2)	f(1)	f(2)	pair
587215.02	174742	708948	1435262	of the
417624.29	129132	540363	1435262	in the
403795.56	16797	16816	18733	Associated Press
387423.99	17287	24135	17564	United States
281913.17	12956	24177	13364	New York
279548.21	12452	17858	13778	All Rights
230483.10	19968	112171	60003	have been
223206.08	66059	258389	1435262	to the
218798.60	31575	447529	110206	to be
211761.70	15576	261695	16816	The Associated
203728.79	17592	95448	60003	has been
200819.84	28748	144059	294607	he said
192063.84	9561	10928	28038	White House
190007.67	17820	69930	110206	will be
173072.45	51336	200524	1435262	on the
161027.74	194	1435262	1435262	the the
157326.20	15799	76338	110206	would be
143998.60	9530	24496	40710	more than
143592.45	10387	19586	89982	did not
142904.19	18933	1435262	24135	the United
137611.41	36922	127433	1435262	at the
135602.11	8555	21820	32378	President Clinton
132831.84	50201	254405	1435262	for the
129884.63	4728	5251	4822	Los Angeles
129815.08	11823	73696	60003	had been
127432.36	8749	127433	11201	at least
124717.59	8950	34048	38146	last year
123596.57	4561	5623	4579	NEW YORK
107720.03	8949	9606	447529	trying to
104395.46	10304	15268	447529	going to
103724.18	4902	24135	5034	United Nations
103113.05	7632	94075	14869	I think
101665.53	31919	132335	143522	from the
101363.56	6307	40505	12238	years ago

Table 11: Sample of highly associated adjacent word pairs from the 1995 Associated Press, using log likelihood ratios. Shown are, from left to right: the value of $-2\log\lambda$; the frequency of the pair f(1,2); the frequency of the first word f(1) and the frequency of the second word f(2). Again, f(1) and f(2) are both greater than 100 for this sample.

$-2 \log \lambda$	f(1,2)	f(1)	f(2)	pair
4988.08	340	14203	2247	most powerful
1959.01	420	607952	2247	a powerful
1336.07	131	24496	2247	more powerful
1093.76	89	7967	2247	most powerful
532.11	57	14183	2247	very powerful
527.49	36	2247	1410	powerful earthquake
438.62	285	1435262	2247	the powerful
363.11	31	2247	3345	powerful lower
309.31	22	1053	2247	politically powerful
293.39	20	2247	776	powerful storms
249.88	30	10575	2247	so powerful
237.09	1	2247	1435262	powerful the
225.00	31	15989	2247	as powerful

Table 12: Possible collocations of *powerful,* along with their log likelihood ratios, from the 1995 Associated Press.

$$I(x, y, z) = \log \frac{P(x, y, z)}{P_I(x, y, z)}$$

where

$$P_I(x, y, z) = P(x)P(y)P(z) + P(x)P(y, z) + P(x, y)P(z)$$

In other words, the mutual information is given, as in the binary case as the log of the probability of the combined event (seeing all three terms), divided by the probability one would expect under an assumption of independence. In the case of a ternary expression, one could find that expression by chance in three ways: all three terms cooccurring by chance; the first term cooccurring by chance with the bigram yz; and the bigram xy occurring by chance with z.

However despite the possibility of such extensions, other approaches have typically been taken to finding multiword collocations. We briefly discuss one such approach here, namely that of Smadja (1993).

Smadja's method starts with a concordance of all instances of a given word w in a corpus. For each w_i found in the context of w, a profile is computed of how often each w_i occurs in each position in a window ranging from five to the left to five to the right of w. A typical example showing the occurrences of *forecast* in a window around *weather,* taken from the 1995 Associated Press corpus are shown in Figure 6. Various conditions are then applied to remove uninteresting words, based on the properties of the profile. For example, it is required that the profile show at least one peak: generally, an interesting collocate of two terms will show some strong preferences for the relative placement of two terms; a flat distribution suggests a pair of words that is only

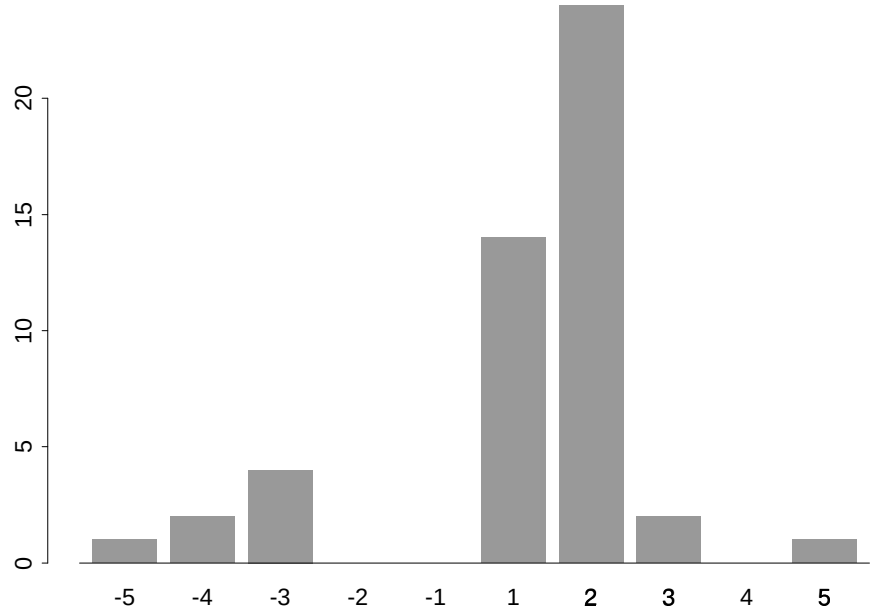


Figure 6: Occurrences of *forecast* in a window around *weather*, computed on the the 1995 Associated Press corpus.

semantically associated, such as *doctor* and *nurse*. Once the interesting two-word collocates are found, further concordances are computed for each pair of words found in the first phase, for each relative position: for example concordances would be computed for *weather* and *forecast* where *forecast* is two words to the right of *weather*. Once again, words that occur in a window of the two words are collected, and those occuring with probability greater than some threshold are kept. Recurring sequences of words are kept as potential collocates. For example, a collocation *blue. . . stocks* discovered during the first phase would be replaced in the second phase by *blue* **chip** *stocks*, since *chip* would occur frequently in this context.

Fung and Wu (1994) have applied Smadja's Xtract system to Chinese.

2.4 Appendix: An Very Brief Note on Chinese Coding Schemes

Apart from UNICODE, there are two coding schemes in common use for Chinese. One is GB (for 國標 *guóbiāo*) used in the Mainland and Singapore, and the other is Big5 (大五碼 *dàwǔmǎ*) used in Taiwan and Hong Kong. (There are actually some slight differences between Taiwan and Hong Kong Big5.) We will be using texts coded in both these schemes.

The differences between the schemes are at three levels. First of all, Big5 is designed for traditional characters whereas GB is designed for the Mainland "simplified" characters. While it would be possible to have, say a GB-coded set of traditional characters (and indeed such sets are sometimes found), most GB-coded character sets have simplified glyphs, and all Big5-coded character sets have traditional glyphs.

Secondly the ordering of the glyphs is based on different principles. In Big5 the glyphs are ordered by radicals and stroke count much as in a traditional Chinese dictionary. In GB they are ordered by the alphabetic order of their pinyin transliteration (as one often finds in dictionaries published on the Mainland.)

Third Big5, at about 13,000 characters, is almost twice as big a character set as GB, which has 6,700, though both are two-byte codes. This is because Big5 uses more of the real estate on the code page. Specifically Big5 is a two-byte code where the first byte always has the high bit set, and the second byte may be set or not. In other words, a Big5 character must match the following bit pattern, where *1* means that the bit is set and *?* means that it may be either set or unset:

1	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

In contrast GB requires the high bit to be set in both bytes:

1	?	?	?	?	?	?	?	1	?	?	?	?	?	?	?

3 Applications

In this section we cover some applications of statistical methods, both "practical" applications as well as more purely linguistic applications. We start with a practical application, namely the problem of word segmentation from Chinese text. In Section 3.2 we consider a couple of areas in the descriptive morphology of Chinese where corpus-based techniques have been applied. (Recall that we already discussed one such case in Section 2.2.1.)

3.1 Segmentation Methods

Chinese word segmentation has attracted a lot of attention. Indeed, it is probably the single most widely addressed problem in the literature on Chinese natural language processing. For all that, though, there are still a number of unsolved problems. A couple of decent, though now somewhat dated literature reviews can be found in (Wang, Su, and Mo, 1990; Wu and Tseng, 1993).

Approaches to Chinese word segmentation fall into three categories:

- Non-stochastic lexicon-based methods.
- Purely statistical methods.
- Methods that combine statistical corpus-based methods with information derived from lexica.

The second category, purely statistical methods, have been fairly limited and are discussed in Section 3.1.1.

Outside of this introduction we will have relatively little to say about the first category, non-stochastic lexicon-based methods, although there have been a fairly large number of these. Two issues distinguish the various proposals. The first concerns how to deal with ambiguities in segmentation. The second concerns the methods used (if any) to extend the lexicon beyond the static list of entries provided by the machine-readable dictionary upon which it is based.

The most popular approach to dealing with segmentation ambiguities is the *maximum matching* method, possibly augmented with further heuristics. This method involves starting at the beginning (or end) of the sentence, finding the longest word starting (ending) at that point, and then repeating the process starting at the next (previous) character until the end (beginning) of the sentence. Papers that use this method or minor variants thereof include (Liang, 1986; Li et al., 1991; Gu and Mao, 1994; Nie, Jin, and Hannan, 1994). The simplest version of the maximum matching algorithm effectively deals with ambiguity by ignoring it, since the method is guaranteed to produce only one segmentation.

Methods which allow multiple segmentations must provide criteria for choosing the best segmentation. Some approaches depended upon some form of constraint satisfication based on syntactic or semantic features (e.g. (Yeh and Lee, 1991), which used a unification-based approach). Gan

(1995) used higher level semantic and pragmatic information to disambiguate such cases as 馬路上生病了 (*mǎ-lù-shàng shēng-bìng le* '(He) got sick on the road' or *mǎ lù-shàng shēng-bìng le* 'The horse got sick on the road.') Still, others depended upon various lexical heuristics: for example Chen and Liu (1992) attempted to balance the length of words in a three-word window, favoring segmentations which give approximately equal length for each word. Methods for expanding the dictionary include, of course, morphological rules, rules for segmenting personal names, as well as numeral sequences, expressions for dates, and so forth (Chen and Liu, 1992; Wang, Li, and Chang, 1992; Chang and Chen, 1993; Nie, Jin, and Hannan, 1994).

Lexical-knowledge-based approaches that include statistical information generally presume that one starts with all possible segmentations of a sentence, and picks the best segmentation from the set of possible segmentations using a probabilistic or cost-based scoring mechanism. Approaches differ in the algorithms used for scoring and selecting the best path, as well as in the amount of contextual information used in the scoring process. The simplest approach involves scoring the various analyses by costs based on word frequency, and picking the lowest cost path; variants of this approach have been described in (Chang, Chen, and Chen, 1991; Chang and Chen, 1993). More complex approaches such as the *relaxation* technique have been applied to this problem (Fan and Tsai, 1988), and more recently adaptive language models of the type used in text compression have been proposed (Teahan et al., 2000). Note that Chang, Chen and Chen (1991), in addition to word-frequency information, include a constraint-satisfication model, so their method is really a hybrid approach. Several papers report the use of part-of-speech information to rank segmentations (Lin, Chiang, and Su, 1993; Peng and Chang, 1993; Chang and Chen, 1993); typically, the probability of a segmentation is multiplied by the probability of the tagging(s) for that segmentation to yield an estimate of the total probability for the analysis.

Statistical methods seem particularly applicable to the problem of unknown word identification, especially for constructions like names, where the linguistic constraints are minimal, and where one therefore wants to know not only that a particular sequence of characters might be a name, but that it is likely to be a name with some probability. Several systems propose statistical methods for handling unknown words (Chang et al., 1992; Lin, Chiang, and Su, 1993; Peng and Chang, 1993). Some of these approaches (e.g. (Lin, Chiang, and Su, 1993)) attempt to identify unknown words, but do not actually tag the words as belonging to one or another class of expression. In other words, one would identify a string as a unit, but not identify the kind of unit it is. This is not always enough. For example if one identifies a hypothetical unit 馬魁乾 as a word, but fails to identify it as a personal name then one is missing critical information that would help one pronounce the final character 乾 correctly: given that this is a name, one is much more likely to pronounce this as *qián* 'universe' than as *gān* 'dry', and thus guess the pronunciation *mǎ kuíqián*. In contrast, if you find the (presumably) nonce compound 馬肉乾 you would probably want to interpret it as 'horse meat jerky' and pronounce it as *mǎròu gān*.

Following (Sproat and Shih, 1990), performance for Chinese segmentation systems is usually reported in terms of the dual measures of precision and recall. (Precision is the number of correct hits divided by the total number of items selected. Recall is number of correct hits divided by

the number of items that *should have been* selected.) It is fairly standard to report precision and recall scores in the mid to high 90% range. Even with the development of various segmentation standards, as discussed in Section 1.3, most work reports results on non-standard corpora, though one is starting to see papers that report results on standardized corpora; e.g. (Hockenmaier and Brew, 1998).

As we noted in (Sproat et al., 1996), the major problem for all segmentation systems remains the coverage afforded by the dictionary and the methods for dealing with unknown words. The dictionary sizes reported in the literature range from 17,000 to 125,000 entries, and it seems reasonable to assume that the coverage of the base dictionary constitutes a major factor in the performance of the various approaches, possibly more important than the particular set of methods used in the segmentation. Furthermore, even the size of the dictionary per se is less important than the appropriateness of the lexicon to a particular test corpus: as Fung and Wu (1994) showed, one can obtain substantially better segmentation by tailoring the lexicon to the corpus to be segmented. More recently, a very promising iterative method for building up a lexicon has been proposed in (Chang and Su, 1997); we will discuss this work in Section 3.1.2.3, along with some other work on unknown words.

3.1.1 Purely Statistical Approaches

So far as we know, the first purely statistical approach to segmentation — i.e., one that makes use of absolutely no dictionary and that basess all of its segmentation decisions on corpus-derived statistics — is (Sproat and Shih, 1990).

The algorithm in that paper was based on the mutual information between characters and worked as follows:

1. For a string of characters $c_1 \ldots c_n$, find the pair of adjacent characters with the largest mutual information greater than some threshold (optimally 2.5), and group these.

2. Iterate 1 until there are no more characters to group.

(The optimal value of 2.5 was the value that approximately maximized precision and recall.) As an example, consider Table 13, which shows the derivation of the segmentation of 我弟弟現在要坐火車回家 *wǒ dìdì xiànzài yào zuò huǒchē̄ huíjiā̄* ‘My brother wants to ride the train home now’.

There was (as is usual) a tradeoff between precision and recall. At the optimal tuning of these, the system was able to get 90% recall and 94% precision. However there are two things to bear in mind about this result. First of all, the results were judged by hand (which can bias judgments), and we were only considering the correctness of two-character words. Second, the test corpus was derived from the training corpus; this is not as much of a cheat as it seems since recall that there were no word boundaries in the training, and all that was trained was mutual information statistics. Still it does at least mean that the statistics are the best one could hope for and that one could only do worse on another corpus.

	我	弟	弟	現	在	要	坐	火	車	回	家
MI	0	10.4	0	4.2	-2.8	0	0	7.3	2.1	4.7	
1	我	弟	弟	現	在	要	坐	火	車	回	家
2	我	[弟	弟]	現	在	要	坐	火	車	回	家
3	我	[弟	弟]	現	在	要	坐	[火	車]	回	家
4	我	[弟	弟]	現	在	要	坐	[火	車]	[回	家]
5	我	[弟	弟]	[現	在]	要	坐	[火	車]	[回	家]

Table 13: Stages in the derivation of *wǒ dìdì xiànzài yào zuò huǒchē̄ huíjiā̄*. The second line shows the mutual information between adjacent characters.

(Sun, Shen, and Tsou, 1998) have proposed an enhancement of the Sproat and Shih method that makes use not only of mutual information, but also other measures of association. In addition they use a t score for determining, in a character sequence *xyz* whether to group *xy z* or *x yz*:

(66) $$t_{x,z(y)} = \frac{p(z|y) - p(y|x)}{\sqrt{var(p(z|y)) - var(p(y|x))}}$$

(We assume, following (Manning and Schütze, 1999, page 165), that the variance of a probability is $p(1-p)$, or effectively just p for small probability. This point is not clarified in (Sun, Shen, and Tsou, 1998).) Further measures are proposed for deciding how to split in longer sequences. They then present an algorithm that combines these various scores. The recall (which they term "accuracy") on a test set is reported as 91.75%.

More recently (Ge, Pratt, and Smyth, 1999) discuss a method based on the Expectation Maximization algorithm (EM). They assume that words are of length between 1 and k characters long, for some reasonable k, such as 4. Probabilities are assigned randomly initially, and the texts are segmented using these probabilities. The word probabilities are then reestimated based on these segmentations, and the text resegmented. The algorithm is iterated until it converges. One interesting aspect of their results is that initially it would seem that their performance is very poor: they report only 66% recall and 72% precision. However, they note that a large percentage of these errors come from common single-character words — presumably things like 的 *de* (relative marker), 是 *shì* 'be', and 個 — *gè* (classifier) that are glommed together with words with which they frequently cooccur; the Sproat and Shih algorithm also had this problem. A simple postprocessing phase to separate these words increases the recall to 98% and the precision to 91%.

3.1.2 Statistically-Aided Approaches

The majority of work on Chinese segmentation over the past few years has been corpus-based, but not *purely* statistical in the sense that either a base dictionary or else a segmented corpus is assumed as part of the training procedure. We describe a few such methods in this section, starting with our earlier work (Sproat et al., 1996) which is based on weighted finite-state transducers.

Since weighted finite-state transducers, as we shall see, offer a uniform way to represent various kinds of lexical information, this method suggests a unified way to think about the problem of Chinese word segmentation.

3.1.2.1 An Approach Using Weighted Finite-State Transducers The statistical method discussed in (Sproat et al., 1996) uses *weighted finite state transducers* (see Section 3.4) to handle Chinese word segmentation. The basic model works as follows:

- Represent dictionary D as Weighted Finite State Transducer, mapping **from** $ChinChar \cup \epsilon$ **to** $PinyinSyllable \cup POS$. (Note that since the primary application of the Sproat et al system was to text-to-speech, the system was not merely a segmenter, but also a transducer mapping from text into phonemic transcription.)
- Weights on word-strings are derived from frequencies of the strings in a 20M character corpus.
- Represent input I as unweighted acceptor over the set $ChinChar$
- Choose segmentation as $BestPath(I \circ D^*)$

An example is given in Figure 15. Here the point is to segment text *ABCD* using a weighted dictionary containing words *D*, *AB*, *CD*, *ABC*, and associated tags *cl*, *vb*, *nc*, *jj*. The selected output groups *AB* and *CD* as words, with their associated tags.

In the application to Chinese text segmentation, we started with a dictionary and estimated the frequency of words in a corpus. (Since the corpus was unsegmented, this involved an iterative training procedure.) Words in the transducer were then weighted with their negative log probability estimates, using the maximum likelihood estimate.

For morphologically derived words (i.e. those not in the original dictionary but morphologically derivable from them) we estimated the probabilities as follows. First for such words that were not in the dictionary, but *were* in the corpus we again used the maximum likelihood estimate. Thus for attested 青蛙們 *qīngwā*+*men* (frog+PL) '(anthropomorphized) frogs', we simply counted the number of occurrences and used the maximum likelihood estimate.

For unattested forms — e.g. 南瓜們 *nánguā*+*men* (pumpkin+PL) '(anthropomorphized) pumpkins' — we used the Good-Turing estimate to compute probability of aggregate unseen members of the class, and simple bigram backoff model used to estimate probability/cost of particular unseen word. Thus:

(67) $P(\text{南瓜們}) = P(unseen(\text{們}))P(\text{南瓜})$

With a dictionary thus constructed and weighted as a WFST segmentation proceeds as in the toy example in Figure 7. For example, given the dictionary fragment in Figure 8 the input in Figure 9 is analyzed by composing it with the transitive closure of the dictionary. This results in a

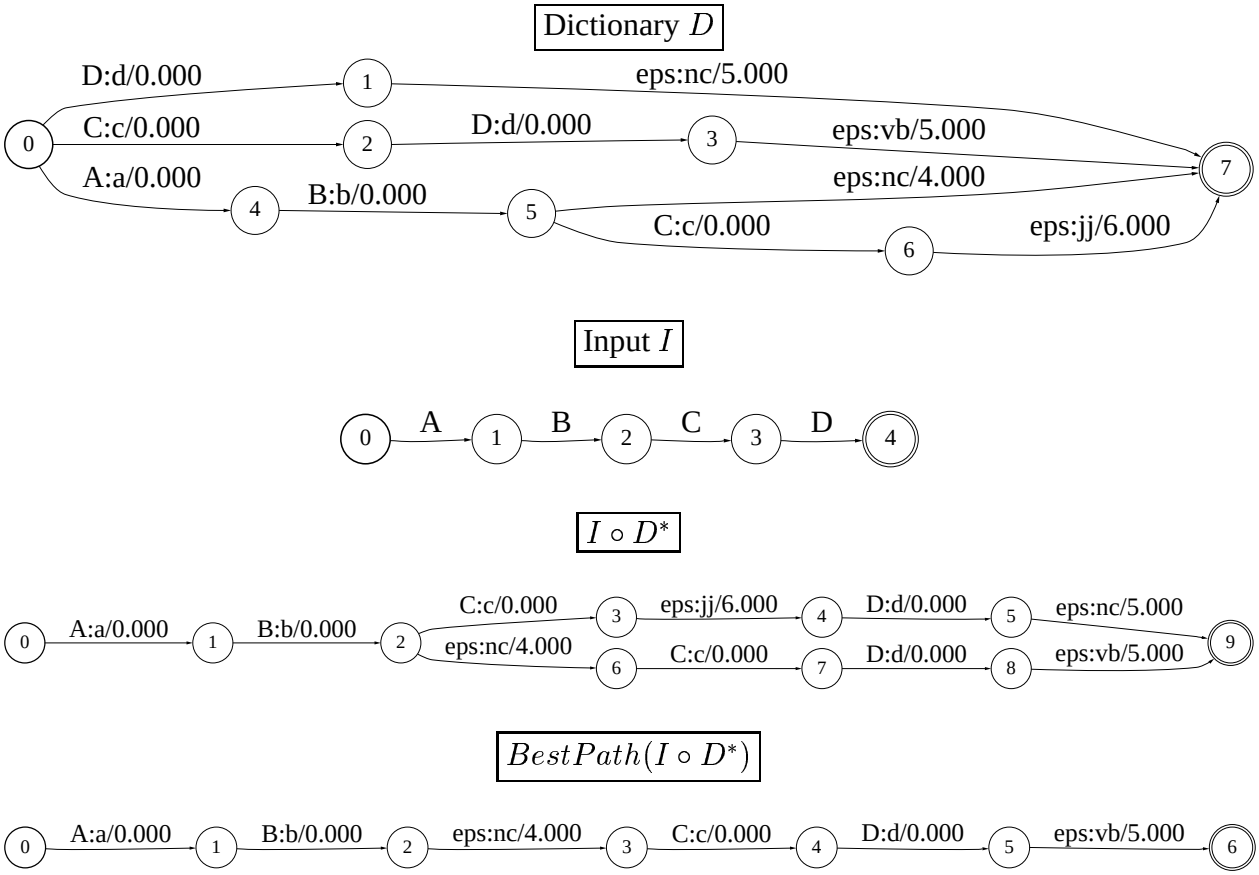


Figure 7: A simple example where the point is to segment text *ABCD* using a weighted dictionary containing words *D*, *AB*, *CD*, *ABC*, and associated tags *cl*, *vb*, *nc*, *jj*. The selected output groups *AB* and *CD* as words, with their associated tags.

lattice of possible analyses, two of the paths of which are shown in Figure 10. The path with the lowest cost (the lower path) wins.

The model just described can be augmented with models of other classes of unknown words. For example, personal names of the form *Family Name+Given Name* can be detected using a finite-state model that allows any family name followed by one or two given-name characters; see also (Chang et al., 1992; Wang, Li, and Chang, 1992). The model used follows the approach of (Chang et al., 1992). For instance we estimate the probability of a name with a single-character family name and two given-name characters as:

(68)

$$p(name|FG_1G_2) \quad = \quad p(|fam| = 1, |giv| = 2) \times p(Fam = F) \times$$

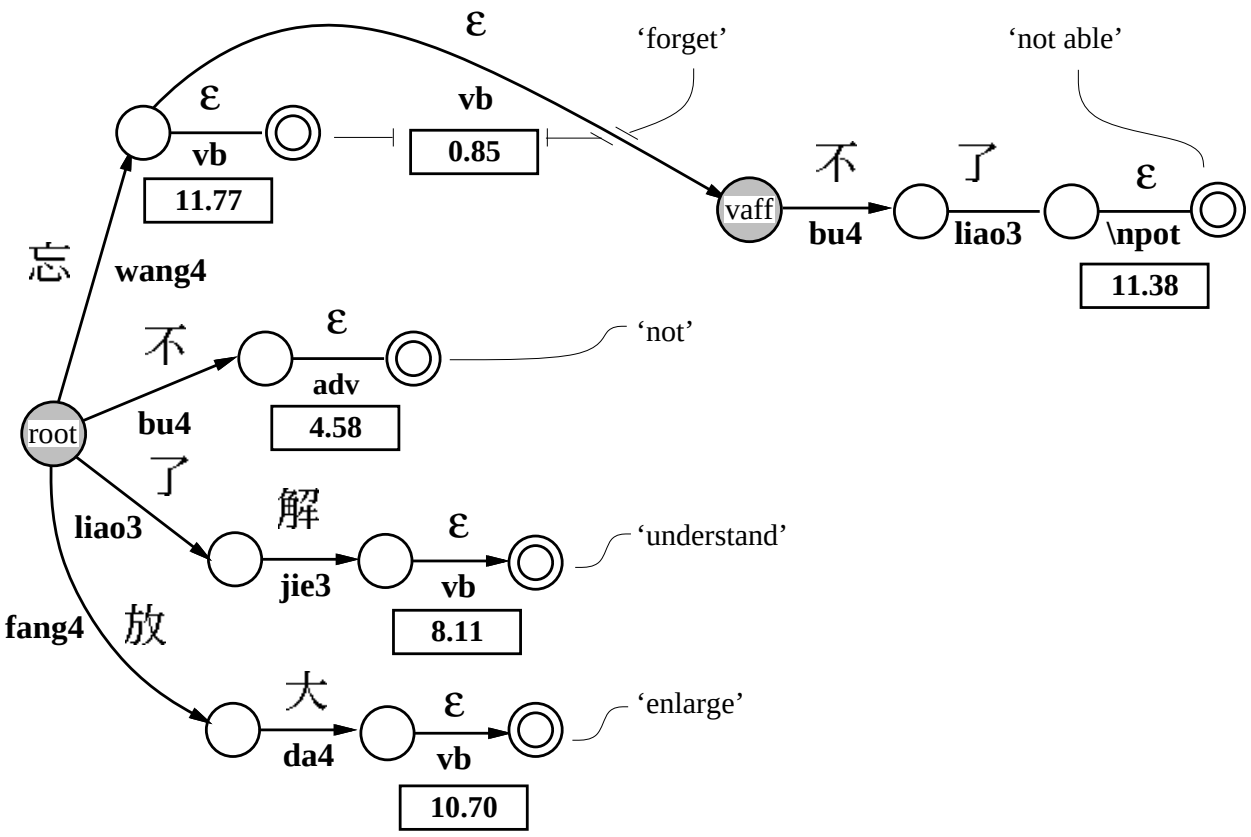


Figure 8: A fragment of the dictionary represented as a WFST.

$$p(Giv_1 = G_1) \times p(Giv_2 = G_2) \times p(name)$$

That is, the probability of the character sequence FG_1G_2 as a name is the product of the probabilities of having a name with a single-character family name and a two-character given name ($p(|fam| = 1, |giv| = 2)$), times the probability of F qua family name ($p(Fam = F)$), times the probability of G_1 qua given name ($p(Giv_1 = G_1)$), times the probability of G_2 qua given name ($p(Giv_2 = G_2)$) times the probability of finding a name ($p(name)$). The final term was estimated from a text corpus, whereas the former were estimated from a list of known names. The model for names is represented as a weighted finite-state transducer that restricts the names to the four legal types (see (42)), with costs on the arcs for any particular name summing to an estimate of the cost of that name.

One parameter that was not well-estimated in the (Sproat et al., 1996) system was the term $p(name)$. This term actually turns out to be critical since it is really the term that tells you how many personal names to expect in a given text. In a list of attendees at a conference, for example, you expect most of the terms to be names, so the estimate should be high. You would presumably

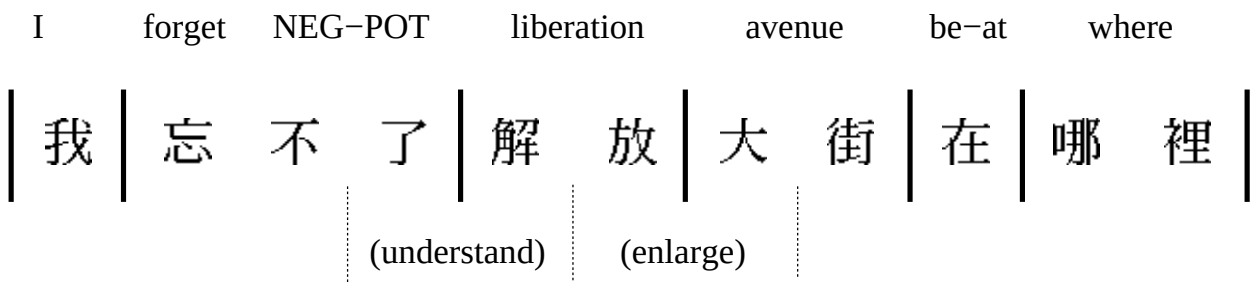


Figure 9: An example input sentence, represented as an FSA.
"I couldn't forget where Liberation Avenue is."

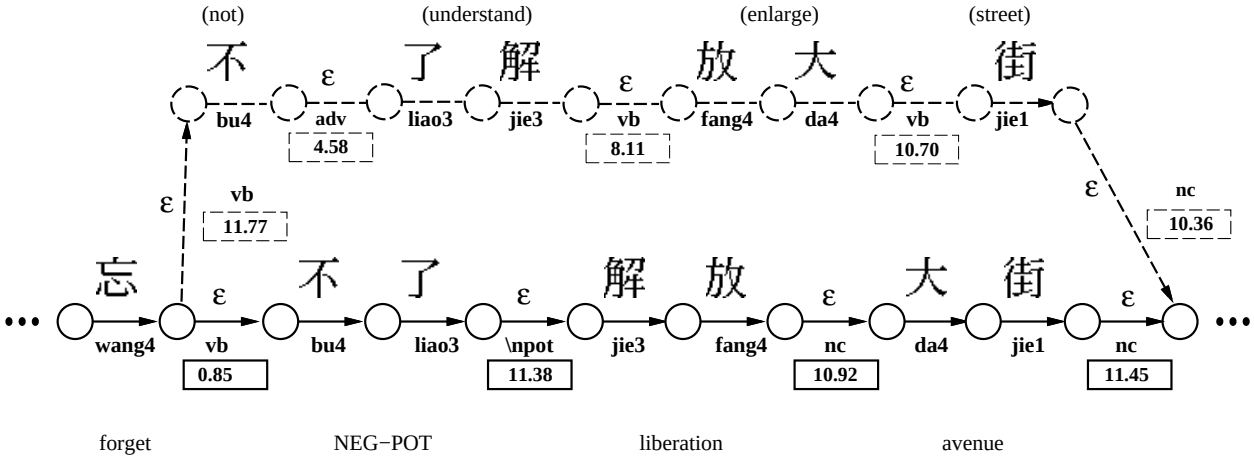


Figure 10: A pair of possible analyses for the sentence in Figure 9 given the dictionary in Figure 8.

want to tune it down, for example, if you knew you were dealing with recipes, where very few names are expected, so that you would avoid classifying 高湯 'broth' as a name.

In a similar fashion foreign transliterated words and names can be handled by observing first of all that only a few characters (± 250) are at all common in transliterations of foreign words; see Table 14, for some of the more frequent of these. In the case of foreign words there are not the formal restrictions that one find in names: transliterated foreign words can have as few as two characters and in principle as many as required to transliterate the name. In practice most are between 3 and 6 characters, so in the implementation we restricted possible names to sequences with lengths within that range.

Since no standard segmented corpus was available at the time this work was done, the evaluation was done by comparing the output of the system with six human judges — 3 from Taiwan (judges "T1"–"T3") and 3 from the Mainland ("M1"–"M3") — who were asked to segment by hand a test corpus of 100 sentences (4,372 characters total). The human segmentations were com-

亞	0.0383
斯	0.0365
拉	0.0334
爾	0.0267
克	0.0218
巴	0.0205
尼	0.0187
利	0.0178
馬	0.0169
阿	0.0151
西	0.0151
蘭	0.0138
羅	0.0134
加	0.0134
里	0.0129
達	0.0125
特	0.0125
德	0.0111
維	0.0102
波	0.0102
哥	0.0089
多	0.0089
布	0.0089
格	0.0076
比	0.0076
倫	0.0071

Table 14: Some characters commonly used in transliterations of foreign names, and their probabilities of occurrence.

Judges	AG	GR	ST	M1	M2	M3	T1	T2	T3
AG		0.70	0.70	0.43	0.42	0.60	0.60	0.62	0.59
GR			0.99	0.62	0.64	0.79	0.82	0.81	0.72
ST				0.64	0.67	0.80	0.84	0.82	0.74
M1					0.77	0.69	0.71	0.69	0.70
M2						0.72	0.73	0.71	0.70
M3							0.89	0.87	0.80
T1								0.88	0.82
T2									0.78

Table 15: Similarity matrix for segmentation judgments

pared with the algorithm just sketched (judge "ST"), as well as:

1. A **greedy** algorithm (or 'maximum matching' algorithm) (judge "GR"): proceed through the sentence, taking the longest match with a dictionary entry at each point.

2. An **anti-greedy** algorithm, (judge "AG"): instead of the longest match, take the shortest match at each point.

We did a pairwise comparison between each of the human and automatic judges, computing the precision and recall in each case, and then computing the arithmetic mean between the two:

(69) $$Similarity = \frac{Precision + Recall}{2}$$

Results are shown in Table 15.

The similarity matrix can be better visualized by computing a classical metric multidimensional scaling (Torgerson, 1958; Becker, Chambers, and Wilks, 1988) on that distance matrix, and plotting the first two most significant dimensions. The result of this is shown in Figure 11. The horizontal axis in this plot represents the most significant dimension, which explains 62% of the variation. In addition to the automatic methods, **AG**, **GR** and **ST**, just discussed, we also added to the plot the values for the current algorithm *using only dictionary entries* (i.e., no productively derived words or names). This is to allow for fair comparison between the statistical method and **GR**, which is also purely dictionary-based. As can be seen, **GR** and this 'pared-down' statistical method perform quite similarly, though the statistical method is still slightly better.

3.1.2.2 Transformation-Based Learning Transformation-based Error-driven Learning (TBL), due to Brill (1993), has been applied to Chinese word segmentation by Palmer (1997) and subsequently by Hockenmaier and Brew (1998). A similar approach using rules learned from corpora is presented in (Chen and Bai, 1998).

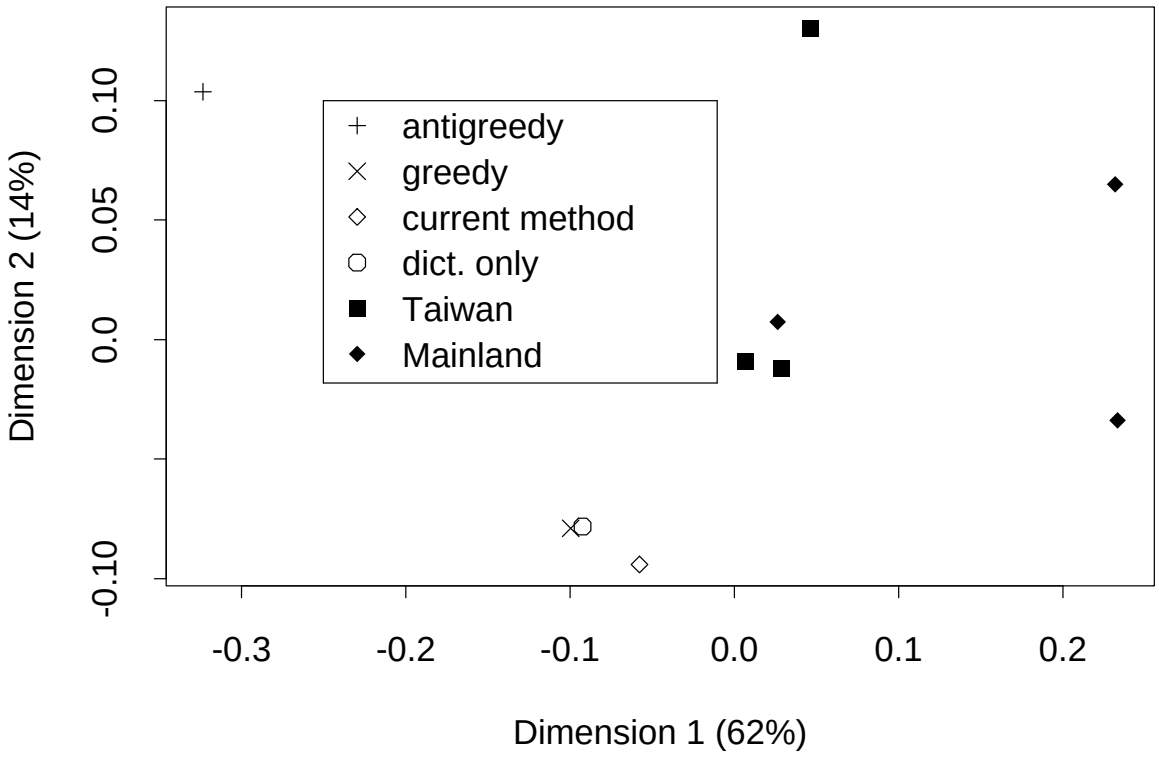


Figure 11: Classical metric multidimensional scaling of distance matrix, showing the two most significant dimensions. The percentage scores on the axis labels represent the amount of variation in the data explained by the dimension in question.

The basic idea behind TBL is simple: one starts with a reference corpus — in the current case a corpus of segmented Chinese, and an initial tagger. The initial tagger is typically simple: in one instance in Palmer's work, for example, the initial tagger simply identified every character as a distinct word. The system is given an inventory of possible transformations. In Palmer's system the inventory was (page 3):

- **Insert** – place a new boundary between two characters
- **Delete** – remove an existing boundary between two characters
- **Slide** – move an existing boundary from its current location between two characters to a location 1, 2, or 3 characters to the left or right

The task is then to iterate, at each stage choosing a transformation that gives the greatest local improvement according to some defined error function. For example, if the current segmenter gives 我 愛 吃 西 瓜 and the reference segmentation is 我 愛 吃 西瓜 *wǒ ài chī xīguā* 'I like to eat watermelon', the system might learn that it should delete a boundary between 西 and 瓜. The process is iterated on the training corpus until there is no further improvement, the initial tagger plus the (ordered) set of transformations learned being the final output of the system. As Palmer notes the system is *weakly statistical*, but not probabilistic, in that it depends upon frequencies in the corpus (a transformation that fixes a lot of problems will in general be preferred over one that fixes a few), but does not rely on probability estimates.

Palmer experimented with various initial conditions including the 'character as word' (CAW) condition mentioned above, and a maximum matching algorithm, and compared the error reduction afforded by the TBL technique. For example, the CAW condition had an initial F-measure[9] of 40.3, which was increased to 78.1 after learning 5,903 transformations. With maximum matching the initial F score was 64.4, increased to 84.9 after acquiring 2,897 transformations.

Hockenmaier and Brew (1998) have extended Palmer's work, proposing somewhat different transformations, and reporting slightly better results in some cases.

3.1.2.3 More on Unknown Words As discussed above, a major problem in Chinese is the problem of unknown words. We discussed a partial solution to this problem in Section 3.1.2.1, which incorporated the previous work on Chinese personal names of (Chang et al., 1992). But this was only a partial solution, and many classes of unknown words were left untreated in the (Sproat et al., 1996) system. In this section we give a brief overview of some other techniques that have been applied to this problem.

Wang, Li and Chang (1992) proposed a method for dealing with names based on recognizing titles (in the case of personal names) and other key words (in the case of, e.g., street names). So a

[9] F is defined in terms of precision P and recall R as follows: $F = \frac{(1+\beta)PR}{\beta P + R}$, where β is a weighting on the precision measure. Palmer used equal weighting, or $\beta = 1$. With perfect precision and recall (100%), $F = 100$.

plausible family name followed by one or two characters, followed by a title such as 先生 *xiānshēng* 'Mr.' or 女士 *nǚshì* 'Ms.' is very likely to be a name. They also propose an "adaptive dynamic word-formation" method whereby a name that has been recognized using a title will automatically be added to the lexicon so that it can be recognized without the title in other portions of the text.

Chen and Chen (2000) propose a method for identifying organization names that uses a lexicon of known organizations along with Chinese texts spidered from the Worldwide Web. The lexicon is first processed to find common suffixes of the terms in the lexicon that are likely to represent productive terms that can be found in other organization names. Thus from 女青年會 *nǚ qīngnián huì* 'Women's Youth Assocation', and other similar terms one can derive the suffix 會 *huì* 'association'. Next texts from the Worldwide Web in particular topic domains are collected and high frequency key words are extracted following (Smadja, 1993) and others. Then, the key words are checked for suffixes that match the organization suffixes collected in the first phase. If the prefix of the name is not an ordinary word (Chinese company names tend not to involve ordinary words), and if the suffix is found with at least n distinct prefixes in the corpus, then the prefix-suffix combination is assumed to be a name. The precision of this method was measured at between 90% and 100% depending upon whether n was set to 2 or 3, respectively. (The recall rate could not be readily measured since the text corpus was too large to know how many distinct names were to be found: one can only assume the recall is low, though, since only 83 correct types were extracted, out of 31,787 distinct financial texts.)

More recently the Chang and Su (1997) have proposed an iterative method for identifying unknown words. They start with a dictionary, augmented with all n-grams (up to a given length) that occur at least five times in the corpus to be segmented. Word probabilities are estimated as the n-gram probabilities from the corpus at hand. The corpus is then segmented using these initial probabilities. The words found in the corpus are then fed into a "Likelihood Ratio Ranking Module", which removes words that are judged to be unlikely. The remaining words are then used to resegment the text, and the process is repeated.

The Likelihood Ratio Ranking Module itself compares an estimate of the likelihood of the n-gram being in the word class, versus the likelihood of its being in the non-word class:

$$\lambda = \frac{f(x|W)p(W)}{f(x|\bar{W})p(\bar{W})}$$

Here, x is a feature vector associated with a given n-gram; $f(x|W)$ and $f(x|\bar{W})$ are, respectively, the density functions for x being in the word class, versus the non-word class; and $p(W)$ and $p(\bar{W})$ are the prior probabilities of words and non-words. The features (x) used by Chang and Su are the (n-way) mutual information between the characters, and the average entropy.[10] The density functions are then modeled as a mixture of multivariate Gaussian distributions (see (Chang and Su, 1997) for details.)

This iterative method is reported to give good results, yielding an 80% recall and 70% precision for new two-character words in one test (Chang and Su, 1997, Figure 5). What is particularly

[10]The entropy of a set of events $x_1 \ldots x_n$ is defined as: $H = -\Sigma_{i=1}^{n} p(x_i) \log(p(x_i))$.

noteworthy is that precision and recall increase in tandem as one increases the number of iterations. This is in contrast to most work on unknown word detection where one typically finds a tradeoff between precision and recall.

Of course, as we pointed out at the beginning of the discussion on segmentation, it is often not enough to identify an unknown word as a word: one frequently wants to know what *kind* of unknown word it is. Unadulterated, the Chang and Su method cannot tell one this.

3.1.3 A Very Short Note on Segmentation for Information Retrieval

An obvious application for Chinese word segmentation would seem to be information retrieval (IR). And indeed, there has been work on Chinese segmentation done with IR in mind. A review can be found in (Wu and Tseng, 1993) and some later work in (Palmer and Burger, 1997).

The problem with this area is that, to our knowledge, people have yet to make a convincing case that word segmentation actually helps much, if at all, in IR. Wu and Tseng discuss a number of issues, including lexical ambiguity — 牛皮 *niúpí* can literally mean 'cow skin' or (as an idiom) 'exaggeration'. But of course that is no different from the case of English: if you use a web search engine on *bass* you will find pages that have to do with music and pages that have to do with fish. Incorrect segmentations will often not make much difference in searches: if I want to search 中華民國 *zhōnghuá mínguó* 'Republic of China', it shouldn't make any difference what segmentation I assume: I presumably am interested in documents that match that literal string.

A genuine problem would be cases where the string one is searching for happens to be a literal substring of some other word that is quite unrelated to the term I am looking for. So 葡萄牙 *pútáoyá* 'Portugal' happens to contain the substring 葡萄 *pútáo* 'grape'. So if one is searching Chinese pages for 葡萄, one should in principle get hits on 葡萄牙 too. So far as we know, though, there have been no results reported on how much of a problem such misanalyses cause in typical IR tasks, such as web searches. Furthermore since most search engines allow one to refine one's query (e.g. "葡萄 AND NOT 牙", and so forth) it is not obvious that one would necessarily see a gain from potentially buggy segmentation.

3.2 Linguistic Studies

3.2.1 Properties of Morphological Constructions

Huang (1999) argues, based on studies of the Academia Sinica Corpus (Chen et al., 1996), that character-based Mutual Information is a good indicator of wordhood in Chinese, suggesting in particular that a minimal mutual information of about 2, used in (Sproat and Shih, 1990) is a good cutoff point for deciding whether a pair of characters constitutes a word.

Huang also argues that the measure of mutual information correlates well with the type of morphological construction one is dealing with. Seemingly monomorphemic words, for example, have a very high mutual information measure (see Table 16) as do what Huang terms (following (Chao,

鞠躬	júgōng	‘to bow’	17.15
躊躇	chóuchú	‘to hesitate’	20.30
蜘蛛	zhīzhū	‘spider’	18.25
霓虹	níhóng	‘neon’	15.66

Table 16: Monomorphemic words and their mutual information, after (Huang, 1999).

母親	mǔqīn	‘mother’	9.78
教會	jiàohuì	‘to teach’	9.63
游泳	yóuyǒng	‘to swim’	11.77

Table 17: Non-versatile polymorphemic words and their mutual information, after (Huang, 1999).

1968)) “non-versatile” polymorphemic words: that is words that are formally polymorphemic, but where the component morphemes are not productive. (Table 17) In contrast, “versatile” polymorphemic words (those with productive components) tend to have a lower mutual information (Table 18). This result is of course expected, in that high mutual information obtains when we have events that are particularly prone to occur with each other, and with nothing else: recall Table 1, where we listed a number of disyllabic morphemes that were written with characters that share a semantic radical. In these pairs the component characters occur only, or almost only, with each other, and they tend to have very high mutual information (Sproat, 2000). In contrast, clear instances of phrases have low mutual information; Table 19. Huang extends the discussion to borrowed words, where, as discussed previously, the characters in question are used purely for their phonetic values. Among characters used in borrowed words, there can be both “versatile” and “non-versatile” components, with the expected effects on mutual information (Table 20).

3.2.2 Analysis of Suoxie

Another area where corpus-based methods have been applied in Chinese morphology is in the analysis of *suoxie* (Huang, Ahrens, and Chen, 1994; Huang, Hong, and Chen, 1994; Chen, 1996); see also Section 1.2.7.1. One important question to be answered with *suoxie* is how speakers decide which portion to pick for a given word: why is 北京大學 *běijīng dàxué* ‘Beijing University’ abbreviated to 北大 *běidà* and not, for instance, 京大 *jīngdà*?

One class of cases that we will dispense with immediately are the suppletive cases such as 滬 *hù*, for 上海 *shànghǎi* ‘Shanghai’ as in 滬杭鐵路 *hù háng tiělù* ‘Shanghai-Hangzhou Railway’. These must presumably be learned. But for the non-suppletive cases, how do speakers in general know which portion to pick for a given word? And how is it that some words seem to allow either portion to be picked: for example 高雄 *gāoxióng* ‘Kaohsiung’ shows up as 高 *gāo* in 高縣 *gāoxiàn* ‘Kaohsiung County’, but as 雄 *xióng* in 雄中 *xióngzhōng* ‘Kaohsiung High School’.

第一	dìyī	‘first’	5.11
免職	miǎnzhí	‘to dismiss’	3.29
唱歌	chànggē	‘to sing’	8.42

Table 18: Versatile polymorphemic words and their mutual information, after (Huang, 1999).

看魚	kàn yú	‘look at fish’	-1.27
打人	dǎ rén	‘to hit people’	-0.91
一第	yī dì	‘one PREFIX’	-4.40
性人	xìng rén	‘-ity human’	-1.44

Table 19: Phrases and non-words.

One possible theory is that uses an “informativeness criterion” in selecting the character in that one picks the character with the more restrictive meaning. Thus for 中學 *zhōngxué* ‘high school’, one would pick 中 *zhōng* ‘middle’ as the most informative character since it distinguishes 中學 from 小學 *xiǎoxué* ‘elementary school’ and 大學*dàxué* ‘university’. But while this works well enough for 中學, it fails to account for cases like 高雄 ‘Kaohsiung’, where as we have seen, in different circumstances one can pick different characters.

A second possibility is to appeal to some kind of “morphological blocking criterion”. On this story, 高雄中學 *gāoxióng zhōngxué* ‘Kaohsiung High School’ must be abbreviated to 雄中 *xióng zhōng*, because 高中*gāozhōng* already exists as a word (‘high school’, itself a *suoxie*). The problem here is how to decide which word should preempt the other.[11]

(Huang, Ahrens, and Chen, 1994; Huang, Hong, and Chen, 1994; Chen, 1996) propose a different model of *suoxie* based on mutual information, where coocurrence of two characters is measured within a five-word window on the left of the second character. The idea is then that for a two-character word AB, to form a *suoxie* compound with X from one element of AB:

1. If A and B are both *associated* with X (e.g., as measured on a corpus with a 5-character window to the left of X) then pick A (the lefthand member) to form AX. As in (Huang, 1999) a pair of characters are assumed to be associated if the mutual information exceeds 2.

2. Otherwise pick the character that is *least* associated with X.

Note that in computing mutual information, instances of the actual *suoxie* are excluded.

The results of a test of this idea run on the Academia Sinica (20 million characters) for Taiwan county names are shown in Table 21. Those cases where both characters are associated with 縣 are given above the first horizontal line in the table. As per the prediction, all of these select the first

[11] Of course, this is a problem for any theory of morphological blocking.

Non-Versatile			
咖啡	kāfēi	‘coffee’	14.86
拷貝	kǎobèi	‘copy’	12.77
Versatile			
伏特	fútè	‘volt’	4.99
攝氏	shèshì	‘Celsius’	9.43

Table 20: Some borrowed words, and their mutual information .

character in the *suoxie* form. Below the second horizontal line are cases where one or fewer of the characters in the county name are associated with 縣, and in these cases the least associated character is picked. The one outlier is 澎湖 ‘Penghu’, where strictly the second character’s association with 縣 is below threshold, and so should be picked, but instead the first character is picked.

One of the lab exercises for this section will reexamine these results and see if they hold up under other conditions.

Chen (1996) suggests an algorithm for identifying the expansions of *suoxie* that also makes use of mutual information, though in a different way from the way it was used in (Huang, Ahrens, and Chen, 1994). Chen considers disyllabic *suoxie* derived from a pair of disyllabic words. For each putative *suoxie AB*, the idea is to find all pairs of disyllabic words where the first contains *A* and the second *B*. So for 幫教 *bāngjiào*, we’d look for words containing 幫 such as 幫助 *bāngzhù* ‘help’ or 黑幫 *hēibāng* ‘gangster’, and for words containing 教 such as 教導 ‘teach’ *jiàodǎo* or 說教 ‘homily’ *shuōjiào*. Chen’s thesis is that a *suoxie* should be formed from collocations where the words are highly associated. Thus the hypothesis is that the best choice for deciding which pair of words is the basis for the *suoxie* is to pick the pair with the highest mutual information. But there is a problem here: while at an intuitive level Chen’s approach may make sense, it isn’t clear what one would expect in practice. In particular, the *suoxie* in question might well occur almost to the exclusion of the full collocation from which it was derived. One sees this in English abbreviations; see (Sproat et al., 2001, page 45). So in any corpus one might find that the *suoxie* 中油 *zhōngyóu* ‘China Oil’ is far more common than its expanded form (中國石油 *zhōngguó shíyóu*. Indeed in the Academia Sinica Balanced Corpus, 中油 occurs 193 times, 中國石油 only 5 times.

3.3 Other Applications

3.3.1 Inferring Word Classes

Chang and Chen (1995) present a method for the automatic classification of Chinese words into a predetermined number of categories. In effect this is an automatic part-of-speech classification system, one where the part-of-speech tags are inferred automatically rather than relying on a dictionary. One application for this kind of technique is *class-based language modeling* in speech

Full	Suoxie	MI(c_1,縣)	MI(c_2,縣)
桃園 Taoyuan	桃縣	4.39	3.32
宜蘭 Yilan	宜縣	3.11	3.86
苗栗 Miaoli	苗縣	4.40	5.37
彰化 Changhua	彰縣	4.48	2.13
雲林 Yunlin	雲縣	3.78	2.13
嘉義 Chiayi	嘉縣	3.49	2.43
澎湖 Penghu	澎縣	3.33	1.92
花蓮 Hualian	花縣	0.95	4.08
高雄 Kaohsiung	高縣	1.33	3.21
屏東 Pingtong	屏縣	1.00	1.29
台北 Taipei	北縣	2.64	0.81
新竹 Hsinchu	竹縣	0.67	0.66
台中 Taichung	中縣	2.64	0.19
南投 Nantou	投縣	0.58	0.29
台南 Tainan	南縣	2.64	0.58
台東 Taitong	東縣	2.64	1.29

Table 21: Analysis of *suoxie* for Taiwan county names, following (Huang, Ahrens, and Chen, 1994; Huang, Hong, and Chen, 1994). The third and fourth columns give, respectively, the mutual information between the first character and *xiàn* 'county', and that between the second character and *xiàn.* Underlined are the characters chosen to form *suoxie* with *xiàn,* and in cases of low mutual information, the lower of the two values in columns three and four.

recognition where it has been argued (e.g (Brown et al., 1992)) that letting the machine infer the classes rather than relying on dictionaries or other human-derived artifacts may result in more robust systems; there is also the additional benefit that automatic techniques can be used in the absence of already existing lexical resources. Chang and Chen also note the additional complication that in Chinese part of speech assignments are often less clear than they are in English, so human judgments on how to classify words can be unreliable.[12]

In Chang and Chen's formulation, the problem is to find a class assignment ϕ for words that maximizes the probability of a corpus. A bigram approximation of this would state that the estimated probability of the text T of length L — $\hat{p}(T)$ — is modeled as the product over each word w_i of the probability of w_i given the inferred class $\phi(w_i)$, along with the probability of $\phi(w_i)$ given

[12]This is a somewhat bizarre argument, though, since what makes Chinese more difficult than, say, English is simply that there is a great deal more lexical ambiguity. Words in Chinese can frequently function in more than one lexical category. But in a given sentence, it is often clear how a word is functioning (e.g. as a noun or a verb). In any case, since automatic classification schemes invariably make the assumption that any given word will only belong to one category, it is hard to see how such schemes can deal with the problem other than by ignoring it.

the previous $\phi(w_{i-1})$:

(70) $\hat{p}(T) = \prod_{i=1}^{L} p(w_i|\phi(w_i))p(\phi(w_i)|\phi(w_{i-1}))$

This is a fairly standard class-based language model approach where the *context* (the previous class(es)) is used to predict the current class, and the current class is then used to predict the current word. The problem is then to optimize the class assignment of words so that $\hat{p}(T)$ is maximized, which Chang and Chen do using a simulated annealing approach. The process is rather slow: to classify a corpus of 355,347 words, with 23,977 word types, into 120 types took 200 CPU hours on a DEC 3000/500 AXP workstation (approximately 20 MFLOPS).

The system was formally evaluated by measuring the perplexity on a testset, but from a linguistic point of view what is more interesting is whether the derived classes make any sense. Clearly some of them do. For example, class 49 seems to be roughly the class of classifiers:

(71) 戶, 大類, 件, 次, 位, 名, 多名, 枋, 段, 段式, 隻, 條, 瓶, 塊, 層樓 ...

Other classes make less sense:

(72) 中山, 中正, 中興, 仁愛, 文學, 水, 牛, 山, 平等, 打開, 民生, 白蘭地 .....

3.4 Appendix: An Overview of Finite-State Transducers

In this section we give an overview of WFST's and the *(weighted) regular relations* that they compute. This is by no means intended to be a complete introduction to this subject. For further and more complete discussion the reader is urged to consult other sources. For ordinary (unweighted) finite-state *acceptors* (FSA's), and the corresponding *regular languages*, any decent introduction to automata theory (Harrison, 1978; Hopcroft and Ullman, 1979; Lewis and Papadimitriou, 1981, for example) will suffice. For transducers and weighted finite-state machines, there is lamentably less material accessible to the non-specialist, but there are a few papers one might recommend. One is (Kaplan and Kay, 1994), whose discussion we draw on to some extent here. More recent is (Mohri, 1997), as well as a few of the references cited therein. Mohri focuses on applications of WFST's to language and speech processing, and efficient algorithms to support these kinds of applications: some of the algorithms that he discusses are the same as are used in our own finite-state toolkit. Finally, one might consult (Sproat, 1992) for a fairly non-technical introduction to FST's as applied in computational morphology and phonology.

3.4.1 Regular Languages and Finite-State Automata

We start with a notion of an alphabet of symbols, where the entire alphabet is conventionally denoted Σ. Also conventional is the use of the symbol ϵ to denote the *empty string*, which is *not* an element of Σ and which is distinct from the *empty set* $\emptyset$. Regular languages are usually defined with an inductive definition along the following lines:

1. $\emptyset$ is a regular language
2. For all symbols $a \in \Sigma \cup \epsilon$, $\{a\}$ is a regular language
3. If L_1, L_2 and L are regular languages, then so are
   (a) $L_1 \cdot L_2$, the *concatenation* of L_1 and L_2: for every $w_1 \in L_1$ and $w_2 \in L_2$, $w_1 w_2 \in L_1 \cdot L_2$
   (b) $L_1 \cup L_2$, the *union* of L_1 and L_2
   (c) L^*, the *Kleene closure* of L. Using L^i to denote L concatenated with itself i times, $L^* = \cup_{i=0}^{\infty} L^i$.

The above definition defines all and only the regular languages. However, there are additional *closure* properties which regular languages observe. We will make use here of the term Σ^*, which denotes the set of all strings (including the empty string) over Σ.[13]

- *Intersection*: if L_1 and L_2 are regular languages then so is $L_1 \cap L_2$.
- *Difference*: if L_1 and L_2 are regular languages then so is $L_1 - L_2$, the set of strings in L_1 that are not in L_2.
- *Complementation*: if L is a regular language, then so is $\Sigma^* - L$, the set of all strings over Σ that are *not* in L. (Of course, complementation is merely a special case of difference.)
- *Reversal*: if L is a regular language, then so is $Rev(L)$, the set of reversals of all strings in L.

Regular languages are defined as sets of strings. They are commonly *notated* as *regular expressions*. And they can be *recognized* by *finite-state automata*. The equivalence of these three objects is a well-known result of automata theory known as Kleene's theorem(s). A finite-state automaton can be defined as follows (see (Harrison, 1978; Hopcroft and Ullman, 1979; Lewis and Papadimitriou, 1981)):

A finite-state automaton is a quintuple $M = (K, s, F, \Sigma, \delta)$ where:

1. K is a finite set of states
2. s is a designated initial state
3. F is a designated set of final states
4. Σ is an alphabet of symbols, and
5. δ is a transition relation from $K \times \Sigma$ to K

[13] It should be stressed that it is not necessary to explicitly state all of these properties: for example, given union and complementation, closure for intersection follows from De Morgan's Laws. We merely list all of these for completeness.

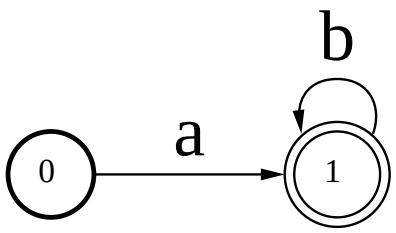


Figure 12: An automaton that computes ab^*. By convention, the heavy-circled state (0) is the initial state, and the double-circled state is the final state.

To illustrate with a simple example, consider the following infinite set of strings: $\{a, ab, abb, abbb \ldots\}$. In other words, the set consisting of a followed by zero or more bs. This can be represented, using fairly standard regular-expression notation, as the compact expression ab^*. And it can be recognized by the finite-state machine depicted in Figure 12. This machine is *deterministic*, meaning that at any state, for a given input symbol, there is exactly one state to which the machine may move. For example, on the input abb, the machine will start in state 0, move on a to state 1, move on b to state 1 again, and repeat this process once more to end in state 1 which, since this state is final, means that the machine has accepted the input. The definition of finite-state automaton above also allows for *non-deterministic* machines, meaning that at a state one may in principle visit more than one next state on a given symbol. As it happens non-deterministic *unweighted automata* can always be determinized — that is, converted into an equivalent deterministic automaton. As a practical matter, an equivalent deterministic automaton may be (much) larger than a non-deterministic automaton, but it will generally be more efficient to operate simply because there is never any need to search more than one path in the machine.[14]

3.4.2 Regular relations and finite-state transducers

One can extend the notion of regular languages to *regular n-relations*, which are sets of n-tuples that can be recursively defined in a parallel manner to the definition of regular languages:

1. $\emptyset$ is a regular n-relation
2. For all symbols $a \in [(\Sigma \cup \epsilon) \times \ldots \times (\Sigma \cup \epsilon)]$, $\{a\}$ is a regular n-relation
3. If R_1, R_2 and R are regular n-relations, then so are
   (a) $R_1 \cdot R_2$, the *(n-way)concatenation* of R_1 and R_2: for every $r_1 \in R_1$ and $r_2 \in R_2$, $r_1 r_2 \in R_1 \cdot R_2$

[14]Strictly speaking, it is not even true that the deterministic machine needs to be larger. See (Mohri, 1994) for discussion of space-efficient representations of deterministic automata.

(b) $R_1 \cup R_2$

(c) R^*, the *n-way Kleene closure* of R.

Regular n-relations can be thought of as *accepting* strings of a relation stated over an m-tuple of symbols, and mapping them to strings of a relation stated over a k-tuple of symbols, where $m+k = n$. We can therefore speak more specifically of $m \times k$-relations. As with regular languages, regular n-relations obey further closure properties:

- *Composition*: if R_1 is a regular $k \times m$-relation and R_2 is a regular $m \times p$-relation, then $R_1 \circ R_2$ is a regular $k \times p$-relation.
- *Reversal*: if R is a regular n-relation, then so is $Rev(R)$.
- *Inversion*: if R is a regular $m \times n$-relation, then R^{-1}, the *inverse* of R, is a regular $n \times m$-relation.

Composition will be explained below.

In most practical applications of n-relations in natural language and speech processing, $n = 2$, with (obviously) $k = m = 1$; in this specific case, we can speak of a relation mapping from strings of one regular language into strings of another. (A notable exception is the work of Kiraz (1999).) For this reason we will speak henceforth simply of *relations*, and have it unambiguously mean *2-relations*. Note that difference, complementation and intersection are omitted from the set of closure properties, since in general regular relations are *not* closed under these operations, though particular subsets of regular relations, including length-preserving (or "same-length") relations, *are* closed under these operations; see (Kaplan and Kay, 1994).

Corresponding to regular relations are n-way regular expressions and *finite-state transducers* (FST's). Rather than give a formal definition of an FST (which can in any event be modeled on the definition of FSA given above), we will illustrate by example. Consider an alphabet $\Sigma = \{a, b, c, d\}$ and a regular relation over that alphabet expressed by the following set: $\{(a, c)$, (ab, cd), (abb, cdd), $(abbb, cddd)$ $\ldots\}$. In other words, the relation consisting of a mapping to c followed by zero or more b's mapping to d. We can represent this compactly by the two-way regular expression *a:c (b:d)**. A finite-state transducer that computes this relation is depicted in Figure 13. By convention, we will refer to the expressions on the lefthand side of the ':' as the input side, and the expressions on the righthand side as the output side. So in Figure 13, the input side is characterizable by the regular expression ab^*, and the output side by the expression cd^*.

As it happens, the transducer in Figure 13 is deterministic, both on the input side and the output side: for any state and any input or output symbol, there is at most one destination state corresponding to that symbol-state pair. But it is important to bear in mind that not all FST's can be determinized. All *acyclic* FST's[15] are certainly determinizable, but for cyclic transducers the

[15] An acyclic automaton or transducer is one in which there are no cycles, meaning that once one leaves any state, there is no path that will lead back to that state.

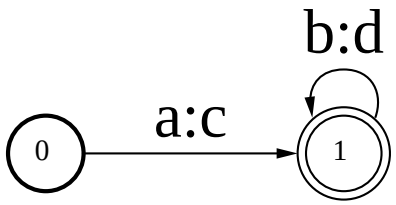


Figure 13: A transducer that computes *a:c (b:d)**.

issue is subtle and complex: the reader is referred to (Mohri, 1997) for extensive discussion of this issue.

The term *composition* has its normal algebraic interpretation: if R_1 and R_2 are regular relations, then the result of applying $R_1 \circ R_2$ to an input expression I is the same as applying first R_1 to I and then applying R_2 to the output of this first application. Consider the two transducers in Figure 14 labeled T_1 and T_2. T_1 computes the relation expressable as *(a:c (b:d)***)* | *((e:g)** *f:h)*. T_2 computes *g:i* ϵ*:j h:k*, where the ϵ*:j* term inserts a j.[16] $T_1 \circ T_2$ computes the trivial relation, *e:i* ϵ*:j f:k*. In this particular case, though both T_1 and T_2 express relations with infinite domains and ranges, the result of composition only maps the string *ef* to *ijk*.

The inverse of a transducer is computed by simply switching the input and output labels. The fact that regular relations are closed under inversion has an important practical consequence for systems based on finite-state transducers, namely that they are fully bidirectional (Kaplan and Kay, 1994). One can for example construct a set of rewrite rules that maps from lexical to surface form; and then invert the resulting transducer to use it to compute possible lexical forms from surface forms.

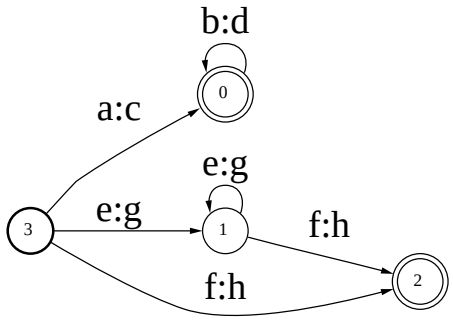
One notion that we will make use of from time to time is the notion of *projection* onto one dimension of a relation. For example, for a 2-way relation R, $\pi_1(R)$ projects R onto the first dimension and $\pi_2(R)$ projects onto the second dimension. (π_1 and π_2 are also said to compute, respectively, the domain and range of the relation.) The analog of the projection operation for 2-way FST's produces an FSA corresponding to one side of the transducer. So the first projection (π_1) of the transducer in Figure 13 is the acceptor in Figure 12.

3.4.3 Weighted regular X and weighted finite-state X

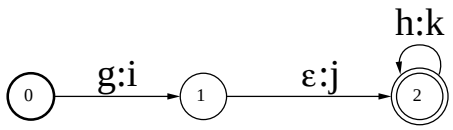
We have so far spoken of automata and transducers where the arcs are labeled with elements from an alphabet. It is also possible to add to these arcs *weights* or *costs*, which may in general be taken from the set of real numbers. Let us consider here the most general case, namely that of a *weighted finite-state transducer* or WFST. An example WFST is given in Figure 15. This WFST accepts any

[16] Here, and elsewhere, we adopt the Unix regular-expression convention whereby '|' denotes disjunction.

T_1



T_2



T_3

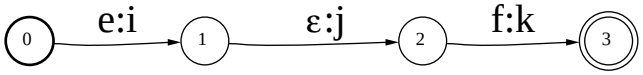


Figure 14: Three transducers, where $T_3 = T_1 \circ T_2$

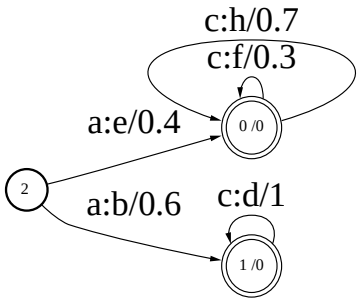


Figure 15: An example of a weighted finite-state transducer. Here, and elsewhere, the number after the '/' is the weight.

string that matches ac^*, and transduces it to several possible outputs. Consider as a concrete case the input acc. The possible outputs for this input are: *bdd*, *eff*, *efh*, *ehf*, *ehh*. The weights allow us to rank these outputs, but the ranking will depend upon how the weights are interpreted. The most common interpretations of the weights in practical applications are:

- As *probabilities*: in this case the weights along a path are *multiplied*, the cost of a particular analysis is the *sum* of the weight of all paths having that analysis, and the best analysis has the *highest* weight.
- As *costs*, or negative log probabilities: in this case the weights along a path are *summed*, the cost of a particular analysis is the *minimum* of the weight of all paths having that analysis, and the best analysis has the *lowest* weight.

Of these two, the cost interpretation is by far the more common (since using real probabilities will lead to underflow problems on large examples), and it is the one that will be assumed here; for the purposes of this discussion, then, *weight* and *cost* have the same meaning, and we will freely interchange these terms. On the cost interpretation, the weights of the possible outputs of acc are as follows: *bdd<2.6>*, *eff<1.0>*, *efh<1.4>*, *ehf<1.4>*, *ehh<1.8>*. Here, and elsewhere we will denote a cost on an expression with a number in angle brackets. Of this set, *eff* has the lowest cost, and is therefore the best analysis. In subsequent discussion, when we refer to the best path, we will always mean the path with the lowest cost, and when we refer to computing the best path, we will presume an algorithm for finding this lowest-cost path.

The closure properties of WFST's are identical to those of FST's. For WFSA's, the closure properties are identical to those of FSA's, except that they are not closed under difference or complementation. On properties for determinizable WFSA's, and algorithms for computing deterministic equivalents see (Mohri, 1997).

4 Corpus Methods for Mandarin Phonology

In casual, conversational speech, many changes occur and the speech stream deviates considerably from the speaker's intended words. This is a nice problem to work on for this section of the course because:

- The task is clearly defined: Casual speech phenomenon can be treated as an operation converting an input phone to an output phone sequence.
- When defined in this way, automatic learning of casual speech rules becomes a problem of string alignment. There are established solutions from works on sequence comparison (Sankoff and Kruskal, 1983).
- There are other automatic rule learning methods such as *Classification And Regression Tree* (CART) (Breiman et al., 1984) that are suitable for the task of learning context-sensitive rules.
- The problem requires us to express linguistic intuition in terms of costs (for string alignment) and factor or attribute coding (for CART). In general, being able to formulate linguistic intuition in mathematical and logical forms helps us to improve or even to design automatic training methods.
- This problem is directly applicable to speech technologies such as automatic speech recognition (ASR) (Liu and Fung, 2000) and text-to-speech synthesis (TTS) (Shih and Sproat, 1996b).

In this section, we will investigate the phenomenon of Mandarin casual conversational speech with a small database. The focus here is to survey the terrain so as to have a good coverage of the problem. In later sections and in lab sections, we will experiment with corpus management methods and automatic learning procedures in order to extend the inquiry to large corpus.

Casual speech phenomenon is at least partially conditioned by prosody (Fosler-Lussier and Morgan, 1999). It is a reasonable hypothesis that many changes occur because the speech tempo is fast and articulatory gestures overlap (Browman and Goldstein, 1990). Therefore it will be useful to think about prosody analysis methods, such as the estimation of duration (van Santen and Olive, 1990; Shih and Ao, 1997) and prosodic strength (Kochanski and Shih, 2001a), and consider how to integrate these information into a model predicting casual speech pronunciation variations.

4.1 Casual Speech Phenomenon

The following example illustrates what might happen in casual speech. The first line and second line give the character and *pinyin* transcription of what the speaker said, respectively. From the pinyin transcription we derive the *ideal* phone sequence, one ascii letter per phone (see Table 22

for the mapping table). This is given on the first indented line. The actual speech output is obtained by labeling the speech phone by phone. This is given on the third indented line. The casual speech phenomenon can be defined as the difference between the input, the *ideal* phone sequence, and the output, the *actual* phone sequence. The mapping can be represented as phone substitution (s), deletion (d) and insertion (i). These operations are given on the second indented line. White space is used when no change occurs.

我想想喔
wo xiang3xiang3 ao0. "Let me think."

```
woxyaGxyaG W*
 s    d  di
wAxyaG ya ?W*
```

A few things happen in this simple example, which point to interesting directions when we ask what might be the conditioning factors of phone changes in casual speech.

First, the word *wo3* "I" is pronounced [wA] (equivalent to pinyin *wei*). We note that the following syllable [xyaG] *xiang* contains a palatal glide [y]. The high, front tongue gesture of [y] may contribute to the fronting and diphthonization of the vowel. If this is the case, we could categorize the change as phone assimilation, one of the most common phenomenon in fast or casual speech. However, note that if [y] is the likely trigger of the [o] $\rightarrow$ [A] assimilation, then the trigger is, in theory, one phone away from the target. We need to keep a mental note of this fact so that later on we can make informed decision on the optimal window size of phone context for automatic rule training. At the same time, keep in mind that in reality, the trigger is probably much closer to the target: fricatives in general tend to show strong co-articulation effect with the following sound and [x] in Mandarin in particular is pronounced with strong palatal coloring. If the articulatory gesture of the palatal sound with high and front tongue position starts a bit earlier than the syllable boundary, it could account for the change of [o]$\rightarrow$[A].

There is a reduplicated verb form *xiangxiang* "think (tentatively)", where the initial consonant [x] of the second syllable is deleted. Since the ideal phone sequences of the two reduplicated syllables are the same (xiang equals xiang), it is tempting to hypothesize that something about the second syllable contributes to the deletion. Is the deletion related to the morphological structure? Could it be that the second syllable is prosodically weak and that the deletion is a lenition process related to the weak position? But also note that the larger phone context of the two [x]'s are different. The first [x] is in the context of [o]__[y], and the second [x] is in the context of [G]__[y]. Is it likely that the preceding context is the cause of the deletion? It is also possible that all of the aforementioned factors are responsible for the disappearing of the [x]. Perhaps the phone context, the morphology structure, and prosody are all relevant factors.

There is one case of insertion in the first example. Generally, insertion isn't a fast speech phenomenon. Intuitively we think that speakers are trying to say less rather than more in order to speak fast. But there are other factors involved. First, note that what is being labeled "fast speech"

is usually not uniformly fast. Conversation speech or casual speech has a wider range of speed than read speech. There are regions where speech is faster and regions where speech is slower than normal.

The questions we should keep in mind is what kind of insertions occur in speech, whether they are predictable, and whether insertion serves a certain function. Here we have a case of glottal stop insertion after [G] (pinyin ng), and in front of a low vowel [W] (pinyin ao). The questions we'd like to ask is whether there are other types of insertion, and with regard to glottal stop insertion, whether it is conditioned by phone context, by lexical property, or both? If the phone context is involved, are both left and right phone contexts relevant? Is the insertion conditioned by phone classes (nasal coda, vowel, or low vowel)?

Given this single example, all we can conclude is that we don't have enough data to support one hypothesis over the other. We need a reasonable sized corpus to answer interesting questions. In a large database, there is also a better chance for us to find varied contexts to estimate the distribution patterns of each of the phenomenon we have seen so far.

Another property of casual speech is variation from utterance to utterance. Any of the described changes may not happen even if the same speaker were to repeat exactly the same materials. One interesting question is to find out how likely each change is, and whether the changes can be predicted by other factors such as speaking rate and prosody. To get a overall picture of the phenomenon, one need an exhaustive analysis of the database, rather than citing anecdotal evidence like we are doing now. In other words, an uninteresting case where nothing happens to [x] is just as important as an *interesting* case where [x] is deleted.

In the following sections, we will test some of the hypotheses on Database K, a small, one and half minute conversational database. We start with a small, manageable database in order to have a detailed documentation on what occurs in the database, their distribution and frequency. The goal here is to understand what happens and what are likely factors that condition the changes. The lessons learned from data analysis will be helpful to find the best strategies for automatic procedures that are necessary to analyze a large corpus.

4.2 Database K and Transcription

Database K is a one and half minute speech database that was recorded during a fieldwork session for an unrelated experiment where the subject was being prompted to talk about specific topics. Only the subject's voice is recorded clearly, and is transcribed and analyzed.

The data representation format is the same as the example shown in Section 4.1.

Table 22 lists the mapping between *pinyin* representations and the phone transcription system when the two systems differ. There are two important considerations to adopt something like the current representation for our purposes: First, we use one ascii symbol to represent each speech sound. This arrangement allows us to simplify string processing operations. Secondly, *pinyin* phone representations may be ambiguous and it is to our advantage to avoid as many unnecessary ambiguities as possible. The consonant representations are very similar to the *pinyin* transcriptions,

Pinyin	(sh)i	(d)e	j(u)	(s)i	er	ou	ei	ai	a(n)
Our Symbol	%	ˆ	U	$	R	O	A	I	@
Pinyin	ao	(d)i(e)	(d)u(o)	yu(e)	sh	ch	zh	(i)n	ng
Our Symbol	W	y	w	Y	S	C	Z	N	G

Table 22: Conversion Chart of Symbols

with the exception of retroflex sounds where we use capital letters *Z, C, S* instead of the two-letter *pinyin* representations *zh, ch, sh*. The vowel mapping admittedly looks weird to human eyes. It grew out of the need to find an unambiguous representation for each vowel sounds with one ascii symbol on the keyboard.

A reasonable reconstruction of the speaker's intention is obtained by transcribing the speech into text. That is, we obtain the listener's impression of what has been said word by word. The first two lines in the following examples are transcriptions of the sentences in characters and in pinyin, respectively. The first indented line show the *ideal* phone string given our reconstruction of what the speaker intended to say. The last indented line gives the *actual* spoken phone string which is obtained by labeling the speech. When one examines the speech stream more carefully and looks for phones, one often finds that the intended phones are mutated or sometimes not even there. The middle indented line show the *operations* that transform the input, or the ideal phone string, to the output, or the spoken phone string, in terms of substitution (s), deletion (d) and insertion (i). We will see in Section 4.4.1 that these three operations are the basic operations used in many string comparison algorithms. They are used to define string distance, which in turn is used to align texts and in automatic rule learning (Kruskal, 1983).

(剛)才說那一大堆話
(gang1c)ai2 shuo1 na4 yi2 da4 dui1 hua4,

ISwonaidadwAhwa*
s d d
ISwonAidad Ah a*

跟你那個
gen1 ni3 nei4ge0,

gˆNninAgˆ*

gˆNninAgˆ*

都沒關係啊
dou1 mei2 guan1xi0 a0.

dOmAgw@Nxia*
s
dOmAgw@Nsia*

喔是呵o3 shi4 h0.

oS%h*
i
?oS%h*

我想想喔
wo xiang3xiang3 ao0.

woxyaGxyaG W*
s d di
wAxyaG ya ?W*

商業中心呀
shang1ye4zhong1xin1 ya0.

SaGyeZoGxiNya*
s
SaGyeZoGsiNya*

美國的商業中心

mei3guo2 de0 shang1ye4 zhong1xin1.

mAgwodˆSaGyeZoGxiN*
s
mAgwoDˆSaGyeZoGxiN*

美國的商業中心你說指
mei3guo2 de0 shang1ye4 zhong1xin1 ni3 shuo1 zhi3,

mAgwodˆSaGyeZoGxiNniSwoZ%*
d s ds
mAgwo ˆSaGyeZoGhiNniS %Z%*

是指地區啊還是城鄉啊
shi4 zhi3 di4qu1 a hai2shi4 cheng2xiang1 a.

S%Z%diqU ahIS%CˆGxyaGa*
i d
S%Z%diqUyahI %CˆGxyaGa*

商業中心
shang1ye4 zhong1xin1.

SaGyeZoGxiN*

SaGyeZoGxiN*

我覺得
wo3 jue2 de,

wojYedˆ*
dss
woj UDˆ*

地區的話我覺得就
di4qu1 de0 hua4 wo3 jue2de jiu4,

diqUdˆhwawojYedˆjyO*
s ds d dd
diqUDˆh Wwoj e jyO*

嗯主要在東西兩岸了

```
en zhu3yao4 zai4 dong1xi1 liang3an4 le.

          ˆNZuyWzIdoGxilyaG @Nlˆ*
         i    d  s   s    di d
         ?ˆNZu WzˆdoGsilya ?@ lˆ*
```

嗯中國的商業中心呢主要在那個大

```
en zhong1guo2 de shang1ye4 zhong1xin1 ne zhu3yao4 zai4 nei4ge da4,

         ˆNZoGgwodˆSaGyeZoGxiNnˆZuyWzInAgˆda*
         s       s         s     ss    s
         eNZoGgwoDˆSaGyeZoGyiNnˆZ%rWzInigˆda*
```

大中城市還有沿海

```
da4zhong1 cheng2shi4 hai2you3 yan2hai3.
         daZoGCˆGS%hIyOyeNhI*
                     ds
         daZoGCˆGS%hI ˆyeNhI*
```

哦我想

```
e wo3 xiang3,
         ˆwoxyaG*

         ˆwoxyaG*
```

哦你是說在中國還是在美國呢

```
e ni3 shi4 shuo1 zai4 zhong1guo2 hai2shi4 zai4 mei3guo2 ne.
         ˆniS%SwozIZoGgwohIS%zImAgwonˆ*
            d  ds     d s  d dd
         ˆni %S ˆzIZoG wahI %  mAgwonˆ*
```

兩邊呢哈

```
liang3bian1 ne ha.
         lyaGbyeNnˆha*
          d       s
         l aGbyeNnaha*
```

哦我覺得在美國呢無所謂

```
e wo3 jue2de zai4 mei3guo2 ne wu2suo3wei4.
         ˆwojYedˆzImAgwonˆ uswowA*
```

```
             d  s       i  d s
        ˆwojYe ˆzˆmAgwonˆwus ovA*
```

因為那個

```
yin1wei4 nei4ge,
         iNwAnAgˆ*
        i ds
        yi vAnAgˆ*
```

美國他這個差別地區差別跟城鄉差別都不太

```
mei3guo2 ta1 zhe4ge cha1bie2 di4qu1 cha1bie2 gen1 cheng2xiang1
cha1bie2 dou1 bu2 tai4,

        mAgwotaZˆgˆCabyediqUCabyeCˆGxyaGCabyedObutI*
                s                   s      ds   s
        mAgwotaZAgˆCabyediqUCabyeCˆGnyaGCab $dOb$tI*
```

不是特別大

```
bu2shi4 te4bie2 da4.

        buS%tˆbyeda*
          s
        bur%tˆbyeda*
```

嗯但是呢

```
en dan4shi4 ne,

        ˆNd@NS%nˆ*
        s    s
        eNd@Nr%nˆ*
```

我可能比較喜歡暖和的地方

```
wo3 ke3neng2 bi3jiao4 xi3huan1 nuan3huo2 de di4fang.

        wokˆnˆGbijyWxihw@Nnw@NhwodˆdifaG*
            d s d  s       dsd   s
        wokˆ ˆNb jyaxihw@Nn a hwoDˆdifaG*
```

要是在中國的話

yao4shi4 zai4 zhong1guo2 de hua4.

yWS%zIZoGgwodˆhwa*
sd s
ya %zIZoGGwodˆhwa*

我就比較傾向於在
wo3 jiu3 bi3jiao4 qing3xiang4 yu2 zai4,

wojyObijyWqiGxyaG UzI*
i
wojyObijyWqiGxyaGyUzI*

那個沿海南方
nei4ge yan2hai3 nan2fang1.

nAgˆyeNhIn@NfaG*

nAgˆyeNhIn@NfaG*

廣州廣州啊深鎮啊
guang3zhou1 guang3zhou1 a shen1zhen4 a.

gwaGZO gwaGZOaSˆNZˆNa*
i
gwaGZOagwaGZOaSˆNZˆNa*

或者是在北京
huo4zhe3 shi4 zai4 bei3jing1.

hwoZˆS%zIbAjiG*
s
hwoZ%S%zIbAjiG*

但是我家在北京啊
dan4shi4 wo3 jia1 zai4 bei3jing1 a1.

```
        d@NS%wojyazIbAjiGa*
         sd      ds
        de r%wojya %bAjiGa*
```

可以經常去看我媽哈

```
ke3yi3 jing1chang2 qu4 kan4 wo4 ma1 ha.

        kˆijiGCaGqUk@Nwomaha*
         s   d    s sdd
        keiji CaGq$ka  omaha*
```

嗯等我有條件的時候

```
en deng3 wo3 you3 tiao2jian4 de shi2hou4.

        ˆNdˆGwoyOtyWjyeNdˆS%hO*
            ds         d
        ˆNdˆG uyOtyWjyeN ˆS%hO*
```

我就想接她出來

```
wo3 jiu4 xiang3 jie1 ta1 chu1lai0.

        wojyOxyaGjyetaCulI*
           ds     d s
        woj $xyaGj ehaCulI*
```

我想孝敬孝敬我媽

```
wo3 xiang3 xiao4jing4 xiao4jing4 wo3 ma.

        woxyaGxyWjiGxyWjiGwoma*
                           ds
        woxyaGxyWjiGxyWjiG uma*
```

哦

e.
        ^*

        ^*

好不容易就是
hao3 bu4 rong2yi4 jiu4shi.

        hWburoG ijyOS%*
          s   i ds
        hWvuroGyij uS%*

可是我現在不行
ke3shi4 wo3 xian4zai4 bu4xing2 .

        kˆS%woxyeNzIbuxiG*
         dd   ds   s
        k %wox iNzIb$xiG*

現在那時間也不允許那個經濟上也不允許
xian4zai4 nei4 shi2jian1 ye3 bu4 yun3xu3 nei4ge
jing1ji4 shang4 ye3 bu4 yun3xu3.

        xyeNzInA S%jyeNyebuUNxUnAgˆjiGjiSaGyebuUNxU*
                i   s   d   s
        xyeNzInAˆS%jyeNyebuUNxUnAgˆjiGniSa yeb$UNxU*

4.3 Analysis of Database K

In this section we will analyze some frequency information trying to (as much as the size of the database allows) quantify these questions: How different is casual or conversation speech from read speech? What kinds of phenomenon occur in conversation, and how frequently? What factors affect the changes? How do we model the changes?

In Database K, there are 708 phones from pinyin/word transcription (The total number of phones from the first indented lines) and 674 phones from manual labeling (The total number of phones from the third indented lines). A simple way to quantify the difference between casual speech and ideal speech is to count the number of operations it takes to convert the input phone sequence to the output phone sequence. In Database K, it takes 113 changes (the number of [s]'s, [d]'s, and [i]'s in the second indented lines) to complete such conversion. In other words, roughly 16% of sounds have been changed one way or the other in this process. Out of the 113 changes, there are 56 phone substitutions, 46 deletions, and 11 insertions. The percentage of change is low compared to the 30% change reported for the English Switchboard speech database (Godfrey, Holliman, and McDaniel, 1992; Greenberg and Fosler-Lussier, 2000).

What are the changes then? We start with an investigation of the most frequently occurring phenomenon. The pinyin transcription of Database K reflect a human transcriber's annotation of word segmentation. Based on these segmentations, we calculated that there are 198 word tokens and 105 unique word types in the database. The most frequent types are given below. The first column shows the frequency. Note that these are frequency count by *pinyin* representation. Homophones will be lumped together, and the same character with different word segmentations will be counted separately. Word segmentation variation also affects the count. But this is sufficient for the current purpose.

```
14      wo
 8      zai
 7      de
 6      a
 5      shi
 5      meiguo
 5      bu
 4      zhongxin
 4      shangye
 4      ne
 4      en
 4      e
```

We also list every change that occurred more than once, again, ranked by frequency. The first column gives the frequency count of the rule. (0 represents null. Following phonological rule writing conventions, 0 occurring in the output of the rule (right hand side of the arrow) indicates

deletion, and 0 occurring in the input of the rule (left hand side of the arrow) indicates insertion. Others are cases of substitution.

```
10        w -> 0
 8        y -> 0
 6        S -> 0
 5        d -> D
 4        d -> 0
 4        0 -> y
 3        x -> s
 3        u -> $
 3        ^ -> e
 3        N -> 0
 3        G -> 0
 3        0 -> ?
 2        z -> 0
 2        w -> v
 2        o -> u
 2        ^ -> 0
 2        Y -> 0
 2        W -> a
 2        S -> r
 2        I -> ^
 2        @ -> a
```

The frequency information suggests that we should pay special attention to [w], [y], [S], [d] and the words *wo*, *zai*, *de*, *shi* (*a* is excluded because it represents sentence final particles or filled pauses, where the identity of the input phone is not clear) . It may not be coincidental that the frequently changed sounds seem to be contained in the frequently occurring words. We now turn to examine several high frequency cases.

4.3.1 A case study of [S]

The sound [S], the voiceless retroflex fricative (pinyin *sh*), occurs 25 times in Database K. Among them, 17 cases are unaffected and 8 cases are changed. Among the changes, there are 6 deletions and 2 substitutions. Both substitutions involve voicing, namely, [S] is substituted by its voiced counterpart [r].

Are these changes governed simply by phone context or by lexical information? We might expect that say, fricatives are less stable, and voiceless sounds tend to get lenited in the intervocalic

context. First, we will see if the distribution of [S] changes with regard to lexical distribution. For this we gather all the [S%] (pinyin *shi*) in the database:

- (城)市 *cheng2shi4*, “city”. (1 occurrence. No change)
- 時候/時間 *shi2hou4/shi2jian1* “time”. (2 occurrence. No change)
- 是: *shi4*, “be” (4 occurrence, 1 change)
  - 是呵 (no change)
  - 是指地區啊 (no change)
  - 或者是在北京 (no change)
  - 你是說 (S delete) [niS%Swo] → [ni%Sˆ]
- Second member of compounds. (7 occurrence, 6 changes)
  - 還是 [hIS%] → [hI%] *hai2shi4* “still” (S delete)
  - 還是 [hIS%] → [hI%] *hai2shi4* “still” (S delete)
  - 不是 [buS%] → [bur%] *bu2shi4* “not” (S → r, voicing)
  - 但是 [d@NS%] → [d@Nr%] *dan4shi4* “but” (S → r, voicing)
  - 但是 [d@NS%] → [der%] *dan4shi4* “but” (S delete)
  - 要是 [yWS%] → [ya%] *yao4shi4* “if” (S delete)
  - 可是 [kˆS%] → [k%] *ke3shi4* “but” (S delete)
  - 就是 *jiu4shi4* “so” (no change, sentence final)

The picture here is relatively clear. This change is lexically governed in the sense that all the changes involve the character 是, “be”. So far, none of the [S%] sound represented by other characters shows any change. Given the convention of Chinese writing system, the usage of distinct characters strongly suggests that these other [S%] came from etymological sources that are distinct from 是. However, we note that 3 cases hardly register on the scale of corpus analysis.

What is more suggestive are the cases where 是 is used as the second member of a compound. In those cases, 6 out of 7 cases involved [S] deletion or lenition in the form of voicing. We also note that these words are sentence connectives such as “still”, “but”, and “if”. It looks like lexical class and the syllable position in the word are good predictors of the sound change. The only exception in this group occurs sentence finally. Perhaps the speed slow down or the word is emphasized? We can check in a large database whether sentence final position influences pronunciation.

4.3.2 A case study of [zI]

We next investigate the sound [I]/[zI]. There 19 instances of the sound [I] in the database, of these, 11 are from the character 在, "located at". We document 4 changes involving the sound [I], all of them are from the word 在. There is one case of syllable deletion ([zI] → 0), two cases of vowel changes to mid vowel ([zI] → [zˆ]) and one case of vowel change to high vowel ([zI] → %). In all three vowel changes, the diphthong is simplified to monophthong.

- 現在 *xian4zai4*, "now". (2 cases. No change)
- 在 *zai4*, "located at" (9 cases, 4 changes)
  - 嗯主要在東西兩岸了 [zI] → [zˆ]
  - 主要在那個 (no change)
  - 哦你是說在中國還是在美國呢 (no change) ... [zI] → 0
  - 哦我覺得在美國呢無所謂 [zI] → [zˆ]
  - 要是在中國的話 (no change)
  - 我就比較傾向於在 (no change)
  - 或者是在北京 (no change)
  - 但是我家在北京啊 [zI] → [%]

Again, we see lexical usage being relevant as a factor conditioning the change. In contrast to the cases of 是 *shi4*, here the monosyllabic word tend to change while the two syllable words are more stable.

We should also note that the vowel changes of在 account for all the changes involving the sound [I].

4.3.3 Case study of *de*

The particle 的 is typically the most frequent character in large modern Chinese text databases. It ranked the third in the small corpus we are using here only because word frequency distribution in a small corpus is strongly affected by the conversation topic. The high frequency of 商業中心 "business center" is a good example.

High frequency words tend to carry less information content, consequently their pronunciations tend to be sloppy and show more variability in casual speech. This is what we find with 的, as well as its variant form 得. In 9 out of 10 cases we find consonant deletion and lenition.

- 的 [dˆ] → [Dˆ]/[ˆ] *de* particle (7 cases, 2 deletions, 4 lenitions, 1 unchanged)
- (覺)得 [dˆ] → [Dˆ]/[ˆ] *de* "feel, think" (3 cases, 2 deletion, 1 lenition)

The 5 cases of lenition [d] → [D] and 4 cases of [d] deletion account for all the [d] deletion and [d] lenition we have in the database.

The consonant changes of [S] and [d], and the vowel change of [I], therefore appear to be lexically conditioned, and the cause are related to the high frequency of 是 "to be", 的, particle and 在 "located at" and their usage as function word.

4.3.4 Case study of [w]

The distribution of deletion and lenition among glides and coda consonants suggests that they are not restricted to specific lexical items. The cause of the change is most likely related to articulatory gestures and the syllable structure.

There are 39 cases of [w] in the database, of which 27 cases are not changed. There are 10 cases of deletion, 2 cases where [w] changes to [v], and also one case of [w] insertion.

```
10        w -> 0
 2        w -> v
 1        0 -> w
```

The sound [w] may be used as an initial consonant, as in the case of 我 [wo] *wo3* "I", 為 [wA] *wei4* "for", or 謂 [wA] *wei4* "say", or it can be used as a post-consonantal glide, as in 國 [gwo] *guo* "country", 說 [Swo] *shuo* "say" and so on. Both cases of [w] → [v] occur on the syllable initial [w], and in particular, on the syllable [wA] and not on [wo]. At the same time, both cases of [wA] in the database show up with the [v] alternation. This is a commonly reported phonological change in northern Mandarin, and we expect to see it given that the speaker is from the north of China. Most likely this change reflects regional accent rather than casual speech phenomenon.

We list the environment where [w] occur in the database:

- Syllable initial glide
  - 我 [wo] *wo3* "I" (14 cases, 3 [w] deletion)
  - 為 [wA] *wei4* "for" (1 case, 1 [w] → [v])
  - 謂 [wA] *wei4* "say" (1 case, 1 [w] → [v])
- Syllable medial glide
  - 國 [gwo] *guo2* "country" (8 cases, no deletion)
  - 話 [hwa] *hua4* "speech" (3 cases, 2 deletion)
  - 說 [Swo] *shuo1* "speak" (3 cases, 2 deletion)
  - 廣(州) [gwaG] *guang3zhou1* "Canton" (2 cases, no change)
  - 所 [swo] *suo3* "with" (1 cases, 1 deletion)

- 暖(和) [nw@N] *nuan3* “warm” (1 case, 1 deletion)
- (暖)和 [hwo] *huo* “warm” (1 case, no change)
- 或(者) [hwo] *huo4zhe3* “or” (1 case, no change)
- 關(係) [gw@N] *guan1xi* “relations” (1 case, no change)
- (喜)歡 [hw@N] *xi3huan1* “like” (1 case, no change)
- 堆 [dwA]→ [dA] *dui* “pile, measure word” (1 case, 1 deletion)

In sum, 7 out of 23 cases of medial [w] glide are deleted, while 3 out of 14 cases of initial [w] glide are deleted. The changes of the syllable medial [w] are distributed in a number of lexical items rather than being restricted to specific words.

Also note that none of the place names (10 occurrences) involve [w] deletion. There are two occurrences of 廣州 *guang3zhou1* “Canton” and 8 occurrences of 國 *guo2* “country”, which are used in contrasting place names 美國 *mei3guo2* “America” and 中國 *zhong1guo2* “China”. Noun phrases tend to be the focal point of the sentence. The information content is high and their occurrences less predictable from context. We expect that under these circumstances, the speaker is motivated to speak clearly.

The one case of [w] insertion occurs before the vowel [u]. The status of this rule is subject to interpretation: One may choose to represent the input as [wu] and therefore consider the current case as no change and treat cases without the [w] glide as cases of deletion. Alternatively, it could be the result of uncertainties in the phoneme’s classification during segmentation. Specifically, it is not a straightforward task to separate a glide from a homoganic vowel. Most likely there is little formant shift from [w] to the following [u], and the primary acoustic cue that could be used to segment a [w] away from the [u] is just a change in amplitude.

4.3.5 Case Study of [y]

We now turn to [y], another case of glide. [y] occurs 45 times in the database. 8 of the [y] are deleted, and 1 is fricated and turns into an [r]. In addition, there are 4 cases of [y] insertion.

```
8       y -> 0
4       0 -> y
1       y -> r
```

We find 2 cases of deletion among 15 syllable initial [y]’s and 6 cases of deletion among 30 syllable medial [y]’s.

The [y] insertion also has exactly the same interpretation problem as [w] insertion.

- Syllable initial glide
  - 商業中心 [ye] *ye* “business center” (5 cases, no change)

- (主)要 [yW] → [W]/[rW] *yao* “mainly” (2 cases, 1 deletion, 1 lenition)
- (還)有 [yO] → [∧] *you* “still” (1 case, 1 deletion with vowel change)
- 要(是) [yW] *yao* “if” (1 case, no deletion, vowel change)
- 沿(海) [yeN] *yan* “along the coast” (1 case, no change)
- 有 [yO] *you* “have” (1 case, no change)
- 也 [ye] *ye* “also” (2 cases, no change)

- Syllable medial glide

  - 想想 [xyaG] “think, tentative” *xiang* (1 case, no change)
  - (想)想 [xyaG] → [ya] *xiang* “think, tentative” (1 case, 1 deletion)
  - 想 [xyaG] “think” *xiang* (3 cases, no change)
  - (城)鄉 [xyaG1] “town” *xiang* (2 cases, no change)
  - (傾)向 [xyaG] “inclined to” *xiang* (1 case, no change)
  - 兩(岸) [lyaG] → [lya] *liang* “both shores” (1 case, no deletion, vowel change)
  - 兩(邊) [lyaG] →[laG] *liang* “both sides” (1 case, 1 deletion)
  - (兩)邊 [byeN] “both sides” *bian* (1 case, no change)
  - (差)別 [bye] → [b$] *bie* “difference” (3 cases, 1 deletion and vowel change)
  - (特)別 [bye] “special” *bie* (1 case, no change)
  - (比)較 [jyW] → [jya] *jiao* “compare” (3 cases, no deletion, 1 vowel change)
  - 就 [jyO] → [j$] *jiu* “then” (3 cases, 1 deletion and vowel change)
  - 就(是) [jyO] → [ju] *jiu* “just so” (1 case, 1 deletion)
  - 家 [jya] *jia* “home” (1 case, no change)
  - 條(件) [tyW] *tiao* “condition” (1 case, no change)
  - (條)件 [jyeN] *jian* “condition” (1 case, no change)
  - 接 [jye] → [je] *jie* “fetch” (1 case, 1 deletion)
  - 孝(敬) [xyW] *xiao* “be filial” (2 cases, no change)
  - 現(在) [xyeN] → [xiN] *xian* “now” (2 cases, 1 deletion)
  - (時)間 [jyeN] *jian* “time” (1 case, no change)

The distribution of [y] deletion/lenition is similar to the pattern of [w] deletion/lenition. Both changes affect a large number of lexical items.

4.3.6 Case of study of nasal coda

Mandarin allows limited syllable structures. Syllable coda consonants are restricted to nasals [N] *n* and [G] *ng*. There are 29 occurrences of [N] and 45 occurrences of [G] in the database. Roughly 10% of them are deleted or lenited.

We list the cases where the coda nasals went through some changes, and omitted cases where no change occurs. It appears that the deletion is not restricted to specific lexical items.

- Alveolar Nasal [N]
  - 因(為) [iN] → [yi] *yin* “because”
  - (兩)岸 [@N] → [?@] *an* “both shores”
  - 暖(和) [nw@N] → [na] *nuan* “warm”
  - 但(是) [d@N] → [de] *dan* “but”
  - 看 [k@N] → [ka] *kan* “see”
  -
- Velar nasal [G]
  - (想)想 [xyaG] → [ya] *xiang* “think tentative”
  - (兩)岸 [lyaG] → [lya] *liang* “two”
  - (可)能 [nˆG] → [ˆN] *neng* “maybe”
  - 經(常) [jiG] → [ji] *jing* “often”
  - (經濟)上 [SaG] → [Sa] *shang* “on”

4.3.7 Summary

We see two distinct patterns of phone changes in casual speech. One is represented by 是, 的 and 在, where the sound changes are restricted to specific lexical items such as high frequency function words. Deletion and lenition are both common. The common sound changes associated with these words typically do not generalize to other lexical items with similar sound combinations.

Another pattern is represented by the changes of glides [y] and [w], and of the nasal codas [N] and [G]. The changes seem to be related to the sound or sound structure, rather than to specific lexical items. Both nasal coda deletion and glide deletion can be viewed as processes which simplify the syllable structure, where complex syllable structures CGV, CVN, and CGVN become CV (C, G, V, N represent consonant, glide, vowel and nasal respectively). In general the percentage of tokens affected is lowerer than the lexically conditioned cases. When a sound change is observed in a broad range of environments, information such as the phone identity and the left and right phone contexts may be useful in predicting their pronunciation variations in casual speech.

We use these case studies to illustrate how one might learn from a corpus, even though the size of the corpus is small. We emphasize that it is important to have a broad variety of environments for each sound, and then analyzing all the data, rather than just reporting interesting but anecdotal observations. Frequency information and a representative list of context quickly provide an overall picture of what is going on, and this information is useful when we try to formulate a model to predict pronunciation variations in a different corpus. We will next investigate methods of automatic rule learning so that we can extend our knowledge base to a large corpus.

4.4 Automatic rule learning

The previous exercise gives some idea of *what* is there to learn. Consider a traditional phonological rule *A changes into B in the context of C if A is a verb* written in the following form:

$A \rightarrow B / \_ C$ If A is a verb

We can separate the learning task into three steps, at least conceptually: The first step is to learn the context free rewrite rule $A \rightarrow B$. The second step is to learn the conditioning context $\_C$. The third step is to learn the restriction that the rule only applies to verbs. In actual implementation, it is possible to combine the steps, especially the conditioning context and the restriction of rule application.

Learning the context free rules is a string alignment problem where we have a sequence of input sounds (the ideal sounds that the speaker intended to say) and a sequence of output sounds (the sounds that the speaker actually produced). The problem is to find the best alignment between the input and the output. We will study the classic string matching algorithm and get familiar with the concept of using weights to tune the output.

There are many machine learning methods that have been used to learn context sensitive rules. We choose the decision tree for this course, for it is conceptually simple, easy to use, and produces acceptable (but certainly not perfect) performance. Learning restrictive conditions such as lexical information and part of speech information is subsumed under decision tree learning, where lexical information, part of speech (POS) information, as well as phone context are used as predictive factors. Prosodic information can be incorporated as well.

4.4.1 Alignment

In previous sections, we considered the linguistic aspect of casual speech phenomenon. This section gives a short introduction to sequence comparison, or string matching, which is the key to finding correspondences between different sequences of codes. The codes are abstract, and could be phones, text, bird songs, or DNA sequences. The casual speech phenomenon we are dealing with is one of the simplest cases because the two sequences we are comparing are very similar. String matching algorithms allow us to obtain the correspondences automatically, and an automatic procedure is necessary to extend our survey to large database. We need to have a basic understanding of the algorithm, in particular, to understand how to use weights to influence the

outcome and to obtain best matches. For this exercise, Joseph Kruskal's article (Kruskal, 1983) *An overview of sequence comparison* is highly recommended.

Here is a summary of the basic concepts from Kurskal's article:

- The problem is to find the best alignment between two sequences.
- There are many different possible ways to align two sequences.
- One can calculate the *distance* between the two sequences if one define what distance means. This distance will depend on how we align the sequences.
- Levenshtein distance is commonly used in string matching algorithms. It counts the number of operations in terms of insertions and deletions (and optionally substitutions) that are needed to convert one string into the other.
- One can assign weights to the operations to influence the outcome.
- The path with the lowest total cost is assumed to be the best solution.
- Dynamic programming is used to implement the search.

We shall start with a few examples and walk through several assumptions that a human might make while matching strings. Consider the following string matching task:

```
abcd
efgh
```

Given strings with equal length and where everything differs, substitution seems to be a good idea.

In contrast, if there are identical units in the two strings, we have a strong intuition that they belong together.

```
woxyaGxyaGW*
wAxyaGya?W*
```

So [w] is aligned with [w] and [W] is aligned with [W]. The desire to match identical units is stronger than the desire to match units from left to right or from right to left. For example, matching [x] with [y] and [y] with [a] does not seem right. This intuition is strong even when we don't know what these units mean.

By matching identical units and maintaining the order of the sequences, we leave gaps that must be accounted for by insertions and deletions. If [y] matches [y], then [x] matches nothing, a case of deletion.

It is not always clear what the *correct* alignment is. Is [G] substituted by [?] or do we have a case of [G] deletion and [?] insertion?

A machine has no capability to differentiate correct and wrong alignments until we define a measure of distance. Without that, some solutions may seem better than others to a human, but it will not be easy to decide which alignment is the best. Without a distance measure, the machine cannot even begin to decide the best alignment. For instance, which of the many possible alignments of the strings below are best? One cannot know without a distance measure.

```
abcd
efghi
```

These intuitions, as well as the common sense that distance cannot be negative, is captured by the *metric axioms* of distance (Kruskal, 1983, page 22). These properties are true for all distance measures, and anything that has these properties is a distance measure. Below, $d(a, b)$ is the distance between a and b.

- Nonnegative property: $d(a, b) >= 0$
- Zero property: $d(a, b) = 0$ if and only if $a = b$
- Symmetry: $d(a, b) = d(b, a)$
- Triangle inequality: $d(a, b) + d(b, c) >= d(a, c)$

Nonnegative property and symmetry expresses the fundamental aspects of distance. Zero property captures the intuition that we prefer matching identical units. Triangle Inequality defines the best path to be the direct and shortest path between two points.

Many string matching algorithms uses *Levenshtein Distance* to calculate the distance between two strings: this distance measure simply counts the numbers of insertions, deletions, and substitutions.

However, one can assign weights to these operations and tune the alignment results. If these three operations are weighted equally, we will get the following as the alignment result:

```
woxyaGxyaGW*
 s    d  s
wAxyaG ya?W*
```

If substitution is weighted more than the sum of 1 deletion and 1 insertion, we will get the following, as a result of globally suppressing substitution.

```
w oxyaGxya GW*
 id    d  id
wA xyaG ya? W*
```

There are inherent ambiguities from the point of view of data interpretation when there are adjacent changes, as illustrated in the following example:

可是我現在不行 "But I can't do it now."

ke3shi4 wo3 xian4zai4 bu4xing2 .

```
kˆS%woxyeNzIbuxiG*
 dd    sd    s
k  %woxi NzIb$xiG*
```

The word *ke3shi4* "but" is reduced to [k%]. Although the proposed analysis as two phone deletions seems reasonable, it is also possible to treat it as one substitution and two deletions, where the substitution could reasonably apply to [ˆ] → [%] by vowel raising, or to [S] → [%] by way of voicing.

Furthermore, [xyeN] of *xian4zai4* "now" is changed to [xiN], which can be treated as either a substitution of [y] → [i] or [e] → [i] and then a deletion of the other vowel, or the whole operation is treated as a merge [ye] → [i]. Again, we can influence the result of rule labeling by assigning weights to the string matching algorithm, or by including operations such as merge.

If our only concern is to build a mapping so that *ke3shi4* will be converted to [k%] and *xian4zai4* will be converted to [xiNzI] in the appropriate contexts, how we label the changes may not impact the eventual outcome. It may not matter whether there are internal ambiguities so long as the cases are treated consistently. But if our goal is to understand the mapping rules or to describe the frequency of phone to phone mapping, then the outcome will be affected by how the mapping relations are labeled. By changing weights of substitutions, deletions and insertions, we can control how the mappings are described, and we can choose the optimal representation (i.e., the simplest) for our purposes.

Problems of this kind exist at all levels and in all types of data analysis, and the management of these kind of problems is particularly important when we want to draw conclusions from a large corpus. It is typically not feasible to investigate each case individually–and even if we do, we sometimes cannot reach a satisfactory solution if the phenomenon is inherently ambiguous. What we can do is to solve a problem as best we can and estimate when and how often the remaining problem occurs.

We have linguistic intuitions that some operations are common, therefore should be encouraged, while some others are rare, therefore discouraged. These intuition often relates to specific sounds or classes of sounds. The rule frequency analysis in the previous section suggests that glide deletion and nasal coda deletion are common. We can favor those rules by assigning low costs to them, thereby obtaining the following results:

```
woxyaGxyaG W*
 s    d  di
wAxyaG ya ?W*
```

We should take care not to violate Triangle Inequality when assigning weights. The following example is a violation of Triangle Inequality in the sense that the most direct route (A $\rightarrow$ C, with a cost of 1.0) is more costly than the indirect route (A $\rightarrow$ B combined with $B \rightarrow$ C, with a cost of 0.4).

- A $\rightarrow$ B costs 0.2: w(A,B)= 0.2
- B $\rightarrow$ C costs 0.2: w(B,C)= 0.2
- A $\rightarrow$ C costs 1.0: w(A,C)= 1.0

Case specific weights can be stored in a matrix of weights as shown in (Kruskal, 1983, page 21). Weights can be estimated from a corpus using various measurements such as weighted frequency.

Database K uses 43 phones (including 0 for deletion and insertion) in the transcription, which means that the matrix of weights has row and column lengths of 43 each, and there are 1849 cells in the matrix. How many of these cells do we have data from Database K for the purpose of estimating weights? Exactly 100 cases, slightly over 5%. Most of the cells represent unlikely sound changes and that is why they don't occur. When the database is small, we have very little information to with which to set the status of an empty cell: does it represents an unlikely sound change, therefore should be assigned a high cost? Or, is it a likely sound change in some context, but we just haven't seen those contexts in the database. If so, perhaps it should be assigned a low cost?

This is a classic problem of data sparsity: a given database covers only a small portion of the possible space. Somehow we need to build a model based on the observations we have from the training set to predict what is going to happen in a new dataset. How well this model generalizes to the new dataset depends on how well the observed cases cover the possible space. When 95% of the possible space is unobserved, can we expect the lessons learned from the training set will generalize to new data?

Increasing the size of the database will help, of course, by increasing the coverage. But there are other effective strategies that we can employ to shrink the space that needs to be covered. As we observed in Database K, sounds in the same category show similar trends. Both the glides [y] and [w] tend to be deleted. If we have no data on another glide [Y], we may guess that the pattern of [Y] deletion will be similar to [w] or [y] deletion. As it turns out, we do have 3 tokens from the same lexical items with the [Y] glide, and it does show a tendency to get deleted.

We can effectively reduce the space of possibilities by combining rows and columns that behave similarly. Sound classes (distinctive features work fine) and sound structure (syllable structure,

word structure etc.) are useful guides which suggest possible ways to combine phones into broader classifications.

The default weights (deletion = insertion = substitution) of the alignment program already work most of the time. We don't really need to fill a 1800 cells matrix of weights to improve accuracy. Assigning weights based on broad phone classes such as 0 (for deletion and insertion), consonant, glide, vowel, and nasal coda requires a matrix with just 25 cells, which is probably sufficient for the current task.

The simple string alignment algorithm as described in (Kruskal, 1983) is implemented in the program *align* (by Dan Lopresti), and a version with weights is implemented in the PM tool (pronunciation model) (by Richard Sproat).

4.4.2 CART

Classification and regression tree (CART) (Breiman et al., 1984) is a very popular machine learning method which belongs to the family of decision tree induction algorithms. We will use CART to learn the conditioning factors of rule application in the lab sections, using the Mandarin stories in the OGI multi-language telephone speech corpus (Cole, Muthusamy, and Oshika, 1992). For this section we recommand reading *Decision Tree Learning*, Chapter 3 of *Machine Learning* (Mitchell, 1997).

We will use a small dataset S, which contains all the [S] samples from Database K, as a test case of decision tree learning.

We have seen in Section 4.3.1 by way of manual analysis that some input [S] are deleted, and the deletion seems to be restricted to lexical items derived from 是 [S%] *shi* "to be". This is most evident when 是 is the second syllable of the word. Many of these words are sentence conjunctions, including "but", "still" and so on. We have not observed any [S] deletion outside of this narrow lexical class. In this section, we will explore how a machine learning algorithm learn to make this observation. It should be emphasized, however, that this is an exercise where the primary goal is to introduce CART tree notation, concept, and algorithm. The attribute coding scheme used here is not at all practical, and the data set is too small. We will return to this issue in Sections 4.4.3 and 4.5.1.3.

4.4.2.1 Data Matrix The first step is to convert the data into a data matrix that can be processed by CART. Needless to say, the data matrix need to contain information that is relevant to the task at hand. Based on our previous observations, we included four coded attributes: P1 (preceding phone), F1 (following phone), POS (part of speech), and Syll (syllable position in the word). The values of P1 and F1 are the identities of the preceding phone and the following phone, respectively. (In later sections we will discuss the problem of this coding scheme and suggest alternatives.) The values of POS are manually coded part-of-speech tags. Syll is represented by two digits, the first digit is the number of syllables in the word and the second digit is the syllable position in the word. Each data point is represented by one line. The input phone is always [S], which therefore does not

need to be coded explicitly. The output phone is given in the last column. There are three output possibilities: [S], [r], and 0 which represents deletion. The non-final columns correspond to coded factors, or attributes. For example, the first line in the data matrix represents an [S] which occurs between [I] and [w]; the POS of the word is "verb"; the word is monosyllabic, so the Syll value is 11. The output value [S] indicates that no change occurs here.

P1	F1	POS	Syll	Output
I	w	verb	11	S
o	%	be	11	S
*	a	noun	41	S
^	a	noun	41	S
^	a	noun	41	S
i	w	verb	11	S
*	%	be	11	S
I	%	conj	22	0
*	a	noun	41	S
^	a	noun	41	S
G	%	noun	22	S
i	%	be	11	0
%	w	verb	11	S
I	%	conj	22	0
u	%	be	22	r
N	%	conj	22	r
W	%	conj	22	0
a	^	noun	21	S
^	%	be	11	S
N	%	conj	22	0
^	%	noun	21	S
O	%	conj	22	S
^	%	conj	22	0
0	%	noun	21	S
i	a	prep	11	S

4.4.2.2 The Tree Figure 16 is an example of a decision tree model which learned the context of [S] deletion from Dataset S . The tree can be used as a prediction model of [S] pronunciation in general. One can simply drop a new data point down the tree, follow the path defined by the coded attribute values, and find the answer represented by the terminal nodes, which are shown in squares.

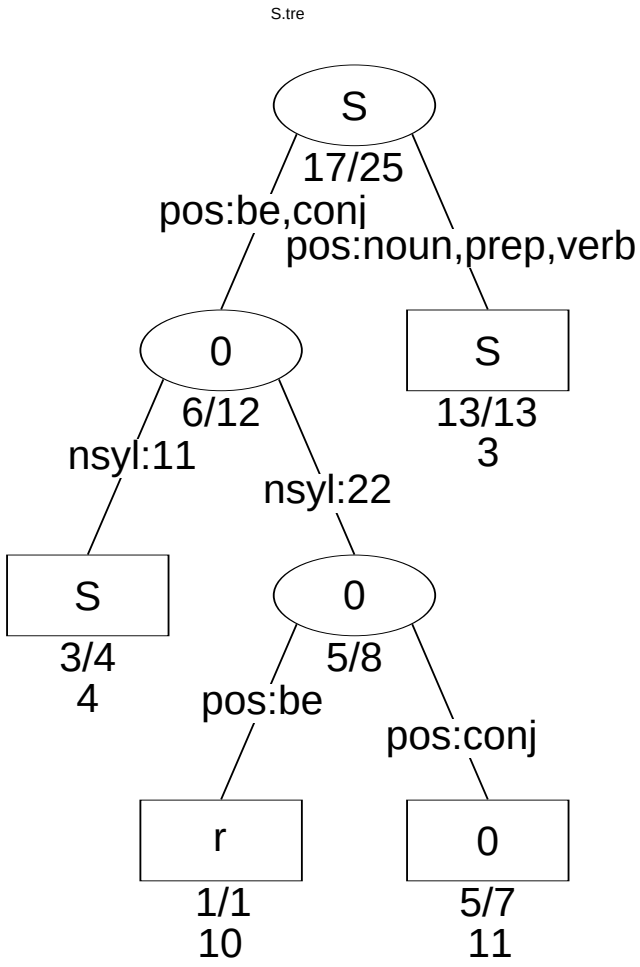


Figure 16: A CART tree model of the pronunciation of [S]

There is a pair of number under each node, which show the population size and the number of samples with the value shown inside the node. For example, 17/25 under the root node reports that the population size is 25, and 17 of them have the value [S] in the output.

The nodes on the path from the root node to the terminal nodes (in squares) should be interpreted as conjunction relations. The left most branch means: If POS is be/conj AND the syllable position is 11, then there is a 75% chance that the output is [S]. Different paths combine in disjunction relations: If POS is be/conj then there is a 50% chance of [S] deletion, OR if POS is noun/prep/verb then [S] remains unchanged.

The machine learning endeavor is successful. This tree reflects previous observations we have made by manual inspection. Part of Speech (POS) tag is the most important attribute determining [S] deletion. If the POS tag is "noun", "preposition", or "verb", then the prediction is that [S] remains unaffected. There is no exception to this observation from the training data. If the POS tag is the verb "be" (是 [S%]) or "conj", then 6 out of 12 cases show [S] deletion. Syllable position in the word further partitions the population under this node: one the one hand, [S] in monosyllabic words (11) are largely unaffected, on the other hand, [S] in the second syllable of a two syllable word (22) is prone to deletion and lenition ([S] $\rightarrow$ [r] by intervocalic voicing).

4.4.2.3 How to Grow a Tree There are many possible way to split the data. How does the decision tree algorithm choose among alternatives? How does it decide to use POS rather than P1 or F1 to split the root node? One early decision trees system, ID3 (Quinlan, 1986) used *information gain* to calculate how well an attribute partitions the data. This measure is widely used in decision tree systems.

Intuitively, we would like to sort the data by their coded attributes so that the outcomes fall into homogeneous bins. For example, if we sort Dataset S by F1, the following phone, we will see that dividing the data into F1 $=$ [%] and F1 $!=$ [%] is kind of helpful in the sense that we get all instances of Outcome $=$ 0 to fall into the bin F1 $=$ [%]. That is an improvement. Instead of saying that there is a 24% chance of [S] deletion, we now can say that there is a 40% chance of [S] deletion if the following phone is [%], and better yet, if the following phone is not [%], we are very sure that nothing happens to [S]. Likewise, we can sort the same dataset by POS or by Syll and see that the divisions help to explain the data. Also revealing is that P1 is NOT helpful. The question is, how can we evaluate the relative merits of P1, F1, POS, Syll numerically so that we know which one works best?

To solve this problem we need to know *entropy*, the measurement of the impurity of a dataset, from information theory (Shannon, 1948). The entropy of a probability distribution P is defined as

$$Entropy(P) \equiv \sum_{i=1}^{c} -p_i log_2(p_i) \tag{1}$$

where c is the number of possible outcomes, and p_i is the probability of the i_{th} outcome. Now,

if we have a dataset D, we can estimate the entropy of the probability distribution from which it is drawn. To do this, we estimate c by the number of different outcomes we observe in D, and we estimate p_i by the proportion of D that gives outcome i. Then, bearing in mind that we will have just an approximate value which is accurate only for large datasets, we can write an estimated entropy for D based on our estimated c and p_i.

We can then compare the entropy of the mother node with the sum of weighted entropy values of the daughter nodes (weighted by the population size) and measure how much we have learned by looking at a given attribute. This is the intuition behind *information gain*, which is defined as:

$$Gain(D, A) \equiv Entropy(D) - \sum_{v \in Values(A)} \frac{|D_v|}{|D|} Entropy(D_v) \quad (2)$$

where D is a dataset, A is the set of attributes, and D_v is the subset of D with attribute value v.

Decision tree algorithms work top-down, starting from the root node, evaluating the information gained by dividing the dataset by various attributes. The best attribute is chosen to split the data, then the same step is applied recursively to divide each daughter node.

At the root node, the population to be divided is dataset S. We can estimate the Entropy of S following Equation 1. For an input phone [S], there are three observed outcomes: [S], [r], or 0 (deletion). Therefore $c = 3$. The population size is 25 and the frequency distributions of the outcomes are 17 [S]'s, 6 deletions, and 2 [r]'s. The entropy of Dataset S is the following (which is really our estimate of the entropy of the probability distribution from which S is drawn).

$$Entropy(S) = -17/25(log_2 17/25) - 6/25(log_2 6/25) - 2/25(log_2 2/25) = 1.16399$$

We now can compare the values of information gain obtained by dividing S by any attribute.

If we divide S into two population F1 = [%] and F1 ! = [%], we get two subsets where the first one includes 3 possible outcomes, 7 [S]'s, 6 deletions, and 2 [r]'s. The estimated entropy of this subset is

$$Entropy(F1_{\%}) = -7/15(log_2 7/15) - 6/15(log_2 6/15) - 2/15(log_2 2/15) = 1.429473$$

The second subset has uniform outcome [S]. The entropy of a uniform population is 0. Plug in the numbers to Equation 2, the information gain of dividing S by $F1_{\%}$ is:

$$Gain(S, F1_{\%}) = 1.16388 - (15/25 * 1.429473 + 0) = 0.3061962$$

If we divide the population by POS = be/conj and POS ! = be/conj, then the first subset has 6 deletions, 4 [S]'s, and 2 [r]'s. The entropy of this population is

$$Entropy(POS_{be,conj}) = -6/12(log_2 6/12) - 4/12(log_2 4/12) - 2/12(log_2 2/12)$$

$$= 1.459148$$

The second subset is again uniformly [S], therefore the entropy is 0. The information gain of this division is:

$$Gain(S, POS_{be,conj}) = 1.16388 - (12/25 * 1.459148 + 0) = 0.463489$$

The POS division works better than the F1 division. Repeating this procedure for all other attributes we find that POS division gives the best scores in terms of information gain. This is the basis for using POS to make the first node division in the [S] tree in Figure 16.

4.4.3 Miscellaneous Issues

In CART tree training, as well as in everyday life, an outcome may be incidentally linked to multiple attributes that are not the true cause of the outcome. There are situations where the most intelligent mind, artificial or not, cannot tease apart the real cause from the incidental ones. When wrong attributes are used to grow a tree, the tree may account for the training data quite well but make poor predictions on new data. This is referred to as the *overfitting* problem.

Figure 17 is an example which shows serious overfitting problem.

To our knowledge (since we have been exposed to a knowledge pool larger than Database K), the change [w] $\rightarrow$ [v] in conditioned by the following phone. In database K, however, the occurrences of F1 = [A] happen to coincide with P1 = [i]/[o] and the model cannot separate the true classifier from the incidental ones. Similar problems occur on many other branches. An overfitted tree like this one is a model built with wrong attributes and it won't generalize to new situation. When we talk about generalizing to a new situation, we are really talking about taking a new sample D' from the underlying probability distribution P. The tree we build will be useful for D' only as far as the entropy estimates we get from D are close to the entropy of P.

There are different techniques to control overfitting. *Cross validation* is an effective method. A dataset is divided into a training set and a validation set. A model learned from the training set is evaluated on the validation set. Some of errors can be put in check using this method. Overfitting problem is less severe in large database where there are more naturally occurring variations.

Figures 17 and 16 both show a different problem of training trees on small database. Note that the training data is so small that the attribute values do not cover the possible attribute space. Many possible preceding phones are not represented in Figure 17. Likewise, many possible POS tags are missing in Figure 16. Needless to say, the trees are not good models of [S] and [w] pronunciations for this reason alone.

Attribute coding is crucial to the success of CART tree learning. The algorithm evaluates available attributes and cannot discover hidden ones. Finding relevant attributes requires a good understanding of the phenomenon at hand. This is one area where data analysis and linguistic knowledge is helpful. Converting knowledge into attributes and attribute values requires some general knowledge building predictive models. We will turn to the topic of prosodic modeling to address some of these questions.

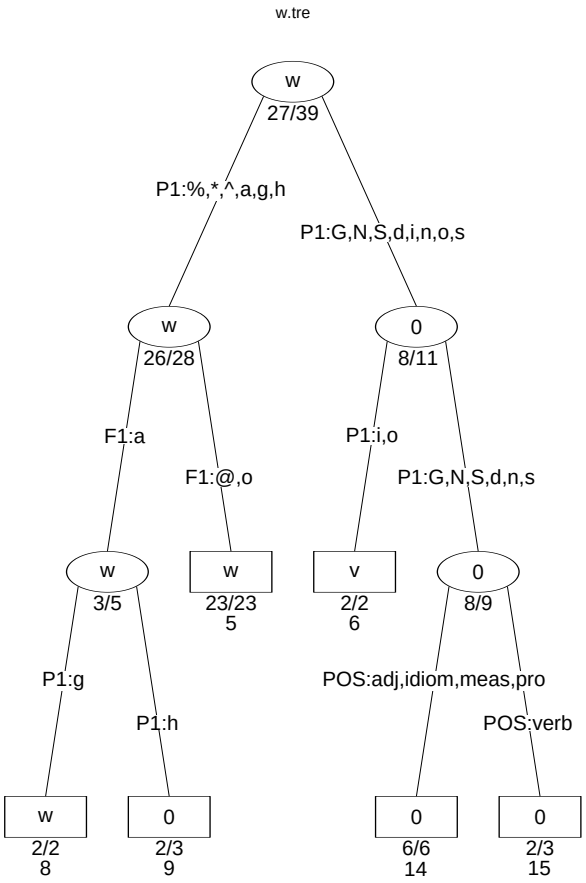


Figure 17: A bad CART tree model of the pronunciation of [w]

4.5 Prosodic Models

Prosodic attributes such as duration and pitch are continuously variable quantities, which are difficult to transcribe just by listening to the speech signal, and that necessitates the use of instrumental measurements of speech in order to gather reliable data. It is not surprising that corpus methods have a long tradition in this field.

This section introduces corpus-based methods for the study of prosody, focusing on the modeling of continuous speech. We will discuss the issues of attribute/factor coding and methods to reduce feature space, which are also relevant to the alignment issue and to CART tree learning. Following the established terminologies in the field, we use attribute/attribute-values in the context of CART tree learning, and factor/factor-levels in the context of prosody modeling (regression analysis).

The ultimate goal of this section is to bring back meaningful representation of prosody that can help us to train pronunciation models. As we will see, sometimes one has to travel far to get a simple answer.

4.5.1 Duration

What aspect of duration information is relevant to pronunciation variations? Researchers in the field reveal their intuition by using the name *fast speech* phenomenon (Dalby, 1984): when speaking speed is fast, we do not have enough time to articulate speech sounds clearly, therefore deletion and assimilation occur. If this intuition is right, then we ought to be able to improve CART tree classification of pronunciation with an extra column of attribute, or factor, coding the speaking rate.

The question is, how do we measure *slow*, *normal* and *fast* objectively?

Slow is slower than normal, and *fast* is faster than normal. So the question can be reduced to *What is normal?*

It is easier to show what it is not. We have a database of 49,624 phones, and the average phone duration is 98.781 msec. Could this be used as the definition of normal? Not really. Using this measure we would classify most of the vowels and affricates as slow and most of the nasals and stops as fast. It does not tell us which vowel and which nasal is more susceptible to pronunciation modification. This measure separates sound classes rather than speaking speed because speech sounds have their intrinsic duration. Under normal circumstances vowels are longer than nasals so it is possible to have a fast vowel being longer than a slow nasal.

Likewise, mean syllable duration doesn't help much. From the same database of 19152 syllables, we obtained the mean syllable duration of 256.35 msec. This measurement separates syllables with complex syllable structure and longer phones from syllables with simple structure and shorter phones. It works better than mean phone duration, but is still far from acceptable as a measure to separate fast and slow syllables.

We can estimate speaking rate only if we have a model that generates a fairly accurate estimate

of normal segmental duration. So now we turn to the question of how to model segmental duration.

4.5.1.1 Factors The task at hand is clearly defined: Given text, we want to predict the duration of each phone. The text, or marked text, will provide the factor or attribute we need in order to predict the duration values. Our training data consists of the same factor codes with corresponding observed duration values, measured from a speech database.

The data matrix is very similar to the one we have used in CART tree training. This is not the only way to represent data, but it is certainly a convenient way.

Basically, this is an extension of a simple controlled experiment. A controlled experiment with one factor is represented as a matrix with two columns, one column is the factor code and the other column is the observed data. The factor and factor levels are chosen a priori by the experiment design. In this exercise, we convert the uncontrolled natural speech data into the format of controlled experiment by posterior interpretation, coding the observed data with as many relevant factors as we can justify.

Therefore, the first step toward building a model is to compile a list of possible factors. We get this list from controlled experiments, data analysis, literatures, intuitions, hypotheses, and asking a question on the LINGUIST list.

There seem to be four groups of factors that affect segmental durations:

- Phone and phone contexts
- Positional factors
  - Phrase final lengthening
  - Word boundary effect
- Emphasis
- Hierarchical structure

Phones have intrinsic duration. Low vowels have longer duration than high vowels, simply because the jaw has longer distance to travel. The manner of articulation has a big effect on consonant duration. It is easy to see that, for example, aspiration and trill both require preparation time to build up air pressure.

Preceding and following phones affect phone duration due to coarticulation effects. If adjacent phones share articulatory gestures, naturally it takes less time to make the transition. Some phone context effects may be phonological in nature, for example, English vowels are longer before a voiced consonant than before a voiceless one even if the consonant voicing distinction is merged in speech (Lisker, 1957; Crystal and House, 1988).

The duration of a phone varies with its position in a sentence. It is likely that duration is used to convey structural information. For example, a phone is typically longer in the sentence final

position (Klatt, 1975; Cooper and Danly, 1981; Edwards, Beckman, and Fletcher, 1991; Berkovits, 1991). A word boundary effect has also been demonstrated perceptually (Cutler and Butterfield, 1990).

Duration reflects how important a word is. Stressed vowels are longer than unstressed vowels, and emphasized words are longer than deaccented words.

Compensatory effects show that higher order structure has an effect on duration. There is a long tradition of study on isochrony (Lehiste, 1972). But the claim that Chinese is a syllable-timed language remains controversial (Campbell and Isard, 1991; van Santen and Shih, 2000)

Most, if not all, of these observations are reflected in the following list of factors, which is a simplified version of the one used in a duration study of Mandarin (Shih and Ao, 1997). Other sources of Mandarin duration studies include (Feng, 1985; Ren, 1985).

1. The current phone
2. The previous phone
3. The following phone
4. The tone
5. Degree of discourse prominence
6. Number of preceding syllables in the word
7. Number of following syllables in the word
8. Number of preceding syllables in the phrase
9. Number of following syllables in the phrase
10. Number of preceding syllables in the utterance
11. Number of following syllables in the utterance
12. Syllable type

Some positional factors make a two-level distinction marking off the initial/final syllable from the rest, some make a three-way distinction of the initial/final syllable, the second one, and the rest. The phrase 商業中心呀 [SaGyeZoGxiNya*] *shang1ye4zhong1xin1 ya0.* “business center huh” is converted into the data matrix below according to this factor list. The last column is the observed duration values in seconds.

```
1    2    3    4   5   6   7   8   9  10  11   12
S    Beg  a    1   0   0   2   0   2   0   2  CVC  0.136
a    S    G    1   0   0   2   0   2   0   2  CVC  0.078
G    a    y    1   0   0   2   0   2   0   2  CVC  0.044
y    G    e    4   0   1   2   1   2   1   2  CV   0.090
e    y    Zcl  4   0   1   2   1   2   1   2  CV   0.074
Zcl  e    o    1   0   2   1   1   2   1   2  CVC  0.058
Z    e    o    1   0   2   1   1   2   1   2  CVC  0.034
o    Z    G    1   0   2   1   1   2   1   2  CVC  0.082
G    o    s    1   0   2   1   1   2   1   2  CVC  0.106
s    G    i    1   0   2   0   1   1   1   1  CVC  0.102
i    s    N    1   0   2   0   1   1   1   1  CVC  0.080
N    i    y    1   0   2   0   1   1   1   1  CVC  0.026
y    N    a    0   0   0   0   1   0   1   0  CV   0.050
a    y    End  0   0   0   0   1   0   1   0  CV   0.156
```

4.5.1.2 Models We use a model to express our view on how the effect of individual factors combine to create the whole picture.

Data analysis is the best way to find out what we should express in a model. If you have no clue, start with simple regression models such as additive models or multiplicative models, or a simple classification model such as CART. If the results are not satisfactory, the problems can lead us to a better model.

In an additive model the effect of each factor adds up. The training data is compared to the following formula, finding the best values for coefficients α_{phone}, $\beta_{PrePhone}$... which minimize the errors. To use this as a predictive model, simply replace the coded factor levels with corresponding coefficients and sum them up. The result is the predicted duration value.

$$Dur = \alpha_{phone} + \beta_{PrePhone} + \gamma_{NextPhone} + ...$$

A multiplicative model fits the data for the following formula, where the terms multiply.

$$Dur = \alpha_{phone} \times \beta_{PrePhone} \times \gamma_{NextPhone} \times ...$$

Suppose a phone is 50 msec long in the unstressed environment and becomes 75 msec in the stressed environment, and another phone is 100 msec unstressed and 125 msec stressed, an additive model works well: the effect of stress is to add 25 msec to the unstressed phone duration. If instead the 100 msec phone becomes 150 msec when stressed, then we need a multiplicative model: the effect of stress is to multiply the unstressed phone duration by 1.5.

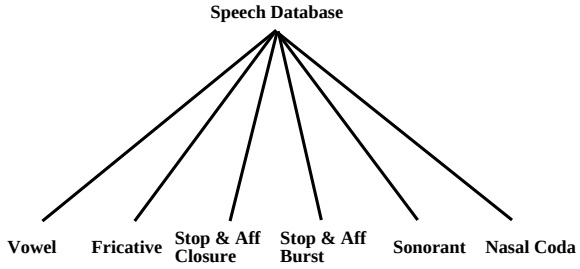


Figure 18: Category tree of Mandarin duration data. Each branch is fitted with a separate model.

A model is an approximation of the observed world. We measure how well competing models work by evaluation the errors, or the discrepancies between the predicted values and the observed values. Because of the overfitting problem, we should evaluate training results on the testing data, rather than on the training data itself.

Duration data works reasonably well with multiplicative models (Allen, Hunnicut, and Klatt, 1987). Most of the Bell Labs Text-to-Speech systems use multiplicative models as well. This is motivated by the observation that lengthening and shortening effects in duration are proportional to phone length. More powerful models, such as sums-of-products models (van Santen, 1993) allow one to express complicated interactions among factors.

4.5.1.3 Managing Feature Space A model will have a good fit if there is a consistent trend in the data that can be captured by the model. Particularly, if we see trend that the effects of each factor levels apply uniformly to all the data points. A dataset that contains all the phones in a language is too complex to work well within this simplistic framework. Consider a few examples. English vowels are lengthened before a voiced stop (Lisker, 1957), but a stop is shortened before another stop. Stress lengthens vowels, but its effect on consonants is less robust. English [p], [t], [k] lose their aspiration after [s], therefore become shorter, but other sound classes are unaffected. These observations suggest that it doesn't make sense to fit a single model for the entire dataset. Minimally, each major sound classes should be fitted separately. Figure 18 shows the data division tree used in the Mandarin duration model (Shih and Ao, 1997). Each branch of the category tree is fitted with a separate multiplicative model.

One would consider further splitting if sound classes interact differently with a major factor. For example, onset consonants and coda consonants are typically modeled separately because they interact differently with neighboring phones and with sentence position: final lengthening has a strong effect on coda consonants. Tense vowels and lax vowels may interact differently with lengthening factors such as stress and sentence final position. If so, one may consider separate models for them.

Splitting a database has its cost. Every split increases the number of models to be fitted and reduces the number of observations available to fit the model, hence the model will be less robust.

One should employ a strategy similar to the one used in the splitting of a CART tree: split the database only if doing so leads to a substantial improvement.

We mentioned in Section 4.4.1 that one can effectively reduce the feature space of the matrix of weights by combining phones into phone classes. This method is applicable here as well. In fact, it is often necessary to collapse some factor levels just to have a solution to the regression model. For example, if one uses stress as well as syllable position in the word as factors to model the duration of a language which stresses word final syllables, and every word is stressed, then there will be no solution because the model cannot separate the effect of word final position from stress. One can eliminate the stress factor or to combine the word final position with another factor level.

Complementary distribution of sounds in a language is another cause of concern. Suppose we want to estimate the effect of the following phone on Mandarin [x] and [S]. It turns out that [x] only occurs before [y], [Y] and [i] while [S] only occurs before sounds other than [y], [Y], and [i], therefore the effect cannot be estimated. Combining [y], [Y], [w] (all the glides) into one level and [i], [U], [u] (all the high vowels) into another level will solve this problem. All cases of complementary distribution in the data matrix should be eliminated by factor level combination.

If we want to model the interaction of factors, the feature space can blow up quickly. If we start by assuming that there is no interaction between factors, the number of parameters for each model is the total number of factor levels plus the number of factors, or roughly 200 parameters as represented in the factor list of Section 4.5.1.1. On the other hand, if we assume that all factors interact, the number of parameters is the product, rather than the sum, of factor levels, which turns out be 183 billions. Finding a database large enough to estimate that many parameters reliably is a challenging task. Even a more modest attempt of modeling phone duration with the tri-phone context (the phone, the preceding phone, and the following phone) translates to a feature space with 97,336 cells. If we assume, in a even simpler model, that it is the phone class, rather than phone identity, of the preceding and following phone that matters and use 10 classes each, the feature space would be reduced to 4600 cells, which is still big but begins to be manageable.

4.5.1.4 Coding Relative Duration With a duration prediction model that takes into account a wide range of factors, we can now estimate whether a section of speech is relatively fast or relatively slow by comparing the observed duration to the predicted duration. We code the relative duration, syllable by syllable, of the sentence 商業中心呀*shang1ye4zhong1xin1 ya0.* "business center huh" as the ratio of the observed duration and the predicted duration. Larger number means slower speaking rate.

Char	pinyin	obs. msec	pred msec	relative duration
商	shang	258	233	1.11
業	ye	164	177	0.93

```
中    zhong    280    204      1.37
心    xin      208    206      1.01
呀    ya       206    172      1.20
```

If speaking rate is indeed one of the factors conditioning sound changes in casual speech, we should see locations of sound change correlated with regions of fast speech. If time allows, we will test this hypothesis in the lab sections.

4.5.2 Intonation

In this section we offer some thoughts on how one might code $f0$ information for the purpose of predicting pronunciation variations.

Converting duration information into codes is relatively easy: the relevant property is speaking speed, and the phone length measurement is reasonably reliable and un-controversial. What is the relevant property of $f0$ contour that is correlated with pronunciation variations, and what is a reasonable measurement?

Browman and Goldstein offered an explanation of *casual speech* phenomenon (Browman and Goldstein, 1990): People spend less effort in area of speech they pay less attention to. The magnitude of articulatory gesture is smaller, and sound changes occur because articulatory gestures blend and overlap.

If we can measure *articulatory effort* from speech signals, then we can make use of this intuition and predict when pronunciation variations is likely to happen. Our recent work on Mandarin prosody modeling (Kochanski and Shih, 2001a) does precisely this. Given $f0$ contours and the location and the identity of lexical tones, we measure how far $f0$ deviates from the expected tone patterns. The result is a numerical measurement of *strength*, or articulatory effort per syllable/tone.

This work is carried out with the prosody description language Stem-ML (Soft TEMplate Mark-up Languages)(Kochanski and Shih, 2001b), which is inspired by Chinese tone data such as Figure 19.

Figure 19 shows the $f0$ track of the phrase反應速度 *fan3ying4 su4du4* "reaction time". The second, third, and fourth syllables all have the same underlying falling tone, but their surface realizations are quite different. The distortion is most dramatic on the second syllable, where the falling tone shows up with a rising shape. This phenomenon happens frequently in conversational data when the syllable is in a prosodically weak position. The resulting tone shape always agrees with the surrounding tonal environment. In other words, the distorted tone shape follows the $f0$ trajectory defined by its neighbors.

It appears that the surface tone shape is the result of an optimization process, where optimality is defined by a balance between the ability to communicate accurately and the effort required to communicate. Specifically, the optimal pitch curve is the one that minimizes the sum of effort plus a scaled error term.

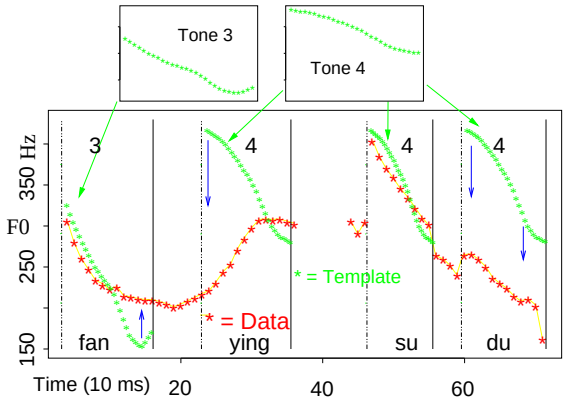


Figure 19: *Tones vs. realization. The upper panels show shapes of tones 3 and 4 taken from carefully read speech in a neutral environment, and the lower panel shows the actual $f0$ contour of a sentence containing those tones. The grey curves show the templates, and the black curve shows the f_0 vs. time data.*

Certainly, when we speak, we wish to be understood, so the speaker must consider the error rate on the speech channel to the listener. Likewise, much of what we do physically is done smoothly, with minimum muscular energy expenditure, so minimizing effort in speech is also a plausible goal.

The error term behaves like a communications error rate: it has its minimum if the prosody exactly matches an ideal tone template, and it increases as the prosody deviates from the template. The choice of template encodes the lexical information carried by the tones. The speaker tries to minimize the deviation, because if it becomes large, the speaker will expect the listener to misclassify the tone and possibly misinterpret the utterance.

The effort expended in speech can be approximated from knowledge about muscle dynamics (Stevens, 1998). Qualitatively, our effort term behaves like the physiological effort: it is zero if muscles are stationary in a neutral position, and increases as motions become faster and stronger. Accordingly, Stem-ML makes one physically motivated assumption. It assumes that f_0 is closely related to muscle tensions. There must then be smooth and predictable connections between neighboring values of f_0 because muscles cannot discontinuously change position. Most muscles cannot respond faster than 150ms, a time which is comparable to the duration of a syllable, so we expect the intonation of neighboring syllables to affect each other. In this sense, our model is an extension of those of (Ö hman, 1967; Fujisaki, 1983; Xu and Sun, 2000).

Effort is ultimately measured in physical units, while the communication error probability is dimensionless, so a scale factor is needed to make the two compatible for addition. This scale factor varies from syllable to syllable, and we identify it with the linguistic strength, or importance of each syllable. If a syllable's strength is large, the Stem-ML optimal pitch contour will closely approximate the tone's template, and the communication error probability will be small. In other

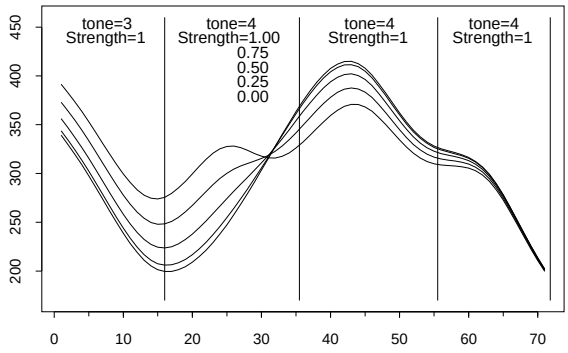


Figure 20: *Variations in $f0$ curves generated from the tone sequence 3-4-4-4, with different strength values on the second syllable.*

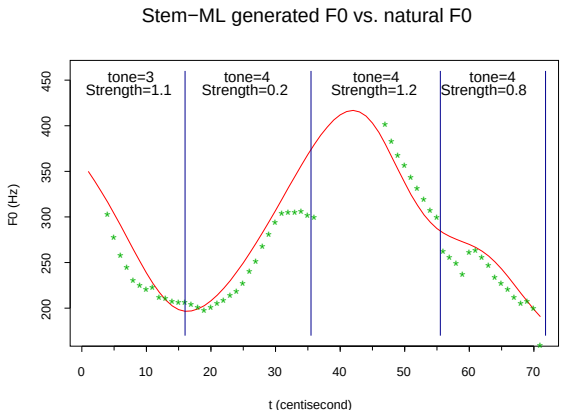


Figure 21: *One strength parameter per syllable is used to generate the $f0$ curve in solid line, which matches the natural $f0$ contour plotted in "*".*

words, a large strength indicates that the speaker is willing to expend the effort to produce precise intonation. On the other hand, if the syllable is unimportant and its strength is small, the produced pitch will be controlled by other factors: neighboring syllables and ease of production. The listener then may not be able to reliably identify the correct tone on that syllable. Presumably, the listener is either able to infer the tone from the surrounding context or the speaker doesn't care if the listener can unambiguously identify the tone.

We then write simple approximations to the effort and error terms, so that the model can be solved efficiently as a set of linear equations.

Figure 20 shows the variations in surface $f0$ generated by the Stem-ML model. All curves are generated from the tone sequence 3-4-4-4, where all tone 4 syllables share the same falling tone template. The strength of the second syllable varies from 0 to 1, while all the other syllables have their strength fixed at 1.

To match a natural $f0$ curve of a Mandarin sentence, we used a set of lexical tone templates and one strength parameter per syllable. Figure 21 shows that the natural $f0$ of 反應速 度shown above can be simulated with strength values 1.1, 0.2, 1.2 and 0.8 for the four syllables. The second syllable is notably weaker than its neighbors, therefore the $f0$ curve over this syllable deviates a lot from what is expected of the lexical tone.

Given $f0$, lexical tone identity and their locations, Stem-ML model can estimate the strength values automatically. This strength is a measure of articulatory effort, which we hope will be useful in locating regions of phone distortion.

4.6 Application to Speech Technologies

Models of pronunciation variations are directly applicable to some branches of speech technologies, such as Text-to-Speech (TTS) systems and automatic speech recognition (ASR) systems.

A TTS system, as the name suggest, convert written text to spoken speech. The system has many components, such as text analysis, duration prediction, and intonation prediction. Text analysis converts written text into phoneme string, duration prediction estimates how long each phoneme is, and intonation module predicts $f0$ contours. All relevant information are send to a speech synthesis component to produce speech output.

The text analysis component simulates the ability of an educated reader. Written text is analyzed and converted into a representation with morphological information, syllable structures, prosodic tags, and phoneme strings. Prediction of pronunciation variation is one area that presents interesting possibilities, but has hardly been explored in the TTS domain. This is primarily because the current systems are still restricted to the *reading* mode.

Proper handling of pronunciation variations will be helpful in generating believable conversation speech, as well as simulating personal characteristics and regional accents.

Speech recognition (ASR) is the reverse process of TTS. It takes speech as input and coverts it into text. TTS is an automatic reader, and ASR is an automatic transcriber.

Pronunciation modeling is a crucial component of an ASR system. An ASR system evaluates acoustic signals against hypotheses of word pronunciation, based on transcriptions and pronunciation models. Multiple pronunciations of a word are annotated or generated so as to cover all the possible pronunciation variations. More complete annotation and good models of pronunciation variations can improve the performance of a ASR system.

Pronunciation modeling of Mandarin casual speech is already integrated in ASR systems. This is one of the research topic at the John Hopkins University Workshop 2000 (Liu and Fung, 2000). Their results and presentations are available from the website http://www.clsp.jhu.edu/ws2000/groups/mcs/.

4.7 Conclusions

This lecture focuses on a single question: the pronunciation variations of Mandarin casual speech. We use this problem as a vehicle to address many methodological issues.

We studied the distribution patterns of Mandarin pronunciation variations in order to build predictive models. To do that we need to understand *what happens*, *why*, *in what context*, *how often* and *why not*.

Automatic methods improves efficiency and enforces objectivity, which make it possible to test hypotheses on a large corpus. We studied string matching algorithms and decision tree learning.

We studied prosody models which give us idea on how to integrate prosody information into pronunciation models.

The most important message of all, I think, is that multi-disciplinary knowledge broadens our view to a linguistic problem. We ventured into many fields and came back with solutions, from statistics, mathematics, physics, information theory, and machine learning. When we solve a linguistic problem, it also has impact to other fields. Speech technologies, for example, are benefited from the understanding of linguistic phenomenon.

References

Allen, J., S. Hunnicut, and D. H. Klatt. 1987. *From text to speech: The MITtalk system*. Cambridge University Press, Cambridge, UK.

Anderson, Stephen. 1992. *A-Morphous Morphology*. Cambridge University Press, Cambridge.

Aronoff, Mark. 1976. *Word Formation in Generative Grammar*. MIT Press, Cambridge, MA.

Baayen, Harald. 1989. *A Corpus-Based Approach to Morphological Productivity: Statistical Analysis and Psycholinguistic Interpretation*. Ph.D. thesis, Free University, Amsterdam.

Baayen, Harald. 1993. On frequency, transparency and productivity. *Yearbook of Morphology 1992*, pages 181–208.

Baayen, Harald. 2001. *Word Frequency Distributions*. Kluwer Academic Publishers, Dordrecht.

Baayen, Harald and Rochelle Lieber. 1991. Productivity and English derivation: a corpus-based study. *Linguistics*, 29:801–843.

Becker, Richard, John Chambers, and Allan Wilks. 1988. *The New S Language*. Wadsworth and Brooks, Pacific Grove.

Berkovits, R. 1991. The effect of speaking rate on evidence for utterance-final lengthening. *Phonetica*, 48:57–66.

Breiman, L., J. H. Friedman, R. A. Olshen, and P. J. Stone. 1984. *Classification and Regression Trees*. Wadsworth International Group, Belmont, CA.

Brill, Eric. 1993. *A Corpus-Based Approach to Language Learning*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.

Browman, Catherine P. and Louis Goldstein. 1990. Tiers in articulatory phonology, with some implications for casual speech. In John Kingston and Mary Beckman, editors, *Papers in Laboratory Phonology I: Between the Grammar and Physics of Speech*. pages 341–376.

Brown, Peter, V. Della Pietra, P. de Souza, Jennifer Lai, and Robert Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguisics*, 18(4):467–479.

Campbell, W.N. and S.D. Isard. 1991. Segment durations in a syllable frame. *Journal of Phonetics*, 19(1):37, 1.

Chang, Chao-Huang and Cheng-Der Chen. 1993. A study on integrating Chinese word segmentation and part-of-speech tagging. *Communications of the Chinese and Oriental Languages Information Processing Society*, 3(2):69–77.

Chang, Chao-Huang and Cheng-Der Chen. 1995. A study on corpus-based classification on chinese words. *Communications of COLIPS*, 5(1-2).

Chang, J.-S., C.-D. Chen, and S.-D. Chen. 1991. Xianzhishi manzu ji jilu zuijiahua de zhongwen duanci fangfa [Chinese word segmentation through constraint satisfaction and statistical optimization]. In *Proceedings of ROCLING IV*, pages 147–165, Taipei. ROCLING.

Chang, Jing-Shin and Keh-Yih Su. 1997. An unsupervised iterative method for chinese new lexicon extraction. *International Journal of Computational Linguistics and Chinese Language Processing*.

Chang, Jyun-Shen, Shun-De Chen, Ying Zheng, Xian-Zhong Liu, and Shu-Jin Ke. 1992. Large-corpus-based methods for Chinese personal name recognition. *Journal of Chinese Information Processing*, 6(3):7–15.

Chao, Yuen-Ren. 1968. *A Grammar of Spoken Chinese*. University of California Press, Berkeley, CA.

Chen, Keh-Jiann and Ming-Hong Bai. 1998. Unknown word detection for Chinese by a corpus-based learning method. *International Journal of Computational Linguistics and Chinese Language Processing*, pages 27–44.

Chen, Keh-Jiann and Chao-jan Chen. 2000. Knowledge extraction for identification of chinese organization names. In *Second Chinese Language Processing Workshop*, Hong Kong.

Chen, Keh-jiann, Chu-Ren Huang, Li-ping Chang, and Hui-Li Hsu. 1996. Sinica corpus: Design methodology for balanced corpora. In B.-S. Park and J.B. Kim, editors, *Proceedings of the 11th Pacific Asia Conference on Language, Information and Computation*, pages 167–176, Seoul. Kyung Hee University.

Chen, Keh-Jiann and Shing-Huan Liu. 1992. Word identification for Mandarin Chinese sentences. In *Proceedings of COLING-92*, pages 101–107. COLING.

Chen, Si-Qing. 1996. The automatic identification and recovery of Chinese acronyms. *Studies in the Linguistics Sciences*, 26(1/2):61–82.

Church, Kenneth Ward and William Gale. 1991. A comparison of the enhanced Good-Turing and deleted estimation methods for estimating probabilities of English bigrams. *Computer Speech and Language*, 5(1):19–54.

Cole, R. A., Y. K. Muthusamy, and B. T. Oshika. 1992. The OGI multi-language telephone speech corpus. In *Proceedings of the International Conference on Spoken Language Processing*, pages 895–898.

Cooper, W. E. and M. Danly. 1981. Segmental and temporal aspects of utterance-final lengthening. *Phonetica*, 38:106–115.

Crystal, T. H. and A. S. House. 1988. Segmental durations in connected speech signals: current results. *JASA*, 83:1553–1573.

Cutler, A. and S. Butterfield. 1990. Durational cues to word boundaries in clear speech. *Speech Communication*, 9(5/6):485, 12.

Dai, John X.-L. 1992. *Chinese Morphology and its Interface with the Syntax*. Ph.D. thesis, The Ohio State University, Columbus, OH.

Dalby, J. M. 1984. *Phonetic structure of fast speech in American English*. Ph.D. thesis, Indiana University.

Di Sciullo, Anna Maria and Edwin Williams. 1987. *On the Definition of Word*. MIT Press, Cambridge, MA.

Dunning, Ted E. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

Edwards, J., M.E. Beckman, and J. Fletcher. 1991. The articulatory kinematics of final lengthening. *JASA*, 89:369–382.

Fan, C.-K. and W.-H. Tsai. 1988. Automatic word identification in Chinese sentences by the relaxation technique. *Computer Processing of Chinese and Oriental Languages*, 4:33–56.

Fano, R. 1961. *Transmission of Information*. MIT Press, Cambridge, MA.

Feng, Long. 1985. Beijinghua yuliu zhong sheng yun diao de shichang (Duration of consonants, vowels, and tones in Beijing Mandarin speech). In *Beijinghua Yuyin Shiyanlu (Acoustics Experiments in Beijing Mandarin)*. Beijing University Press, pages 131–195.

Fosler-Lussier, Eric and Nelson Morgan. 1999. Effects of speaker rate and word frequency on pronunciations in conversational speech. *Speech Communication*, 29:137–158.

Fujisaki, Hiroya. 1983. Dynamic characteristics of voice fundamental frequency in speech and singing. In P. F. MacNeilage, editor, *The Production of Speech*. Springer-Verlag, pages 39–55.

Fung, Pascale and Dekai Wu. 1994. Statistical augmentation of a chinese machine-readable dictionary. In *WVLC-94: Second Annual Workshop on Very Large Corpora*.

Gan, Kok Wee. 1995. *Integrating Word Boundary Identification with Sentence Understanding [?]*. Ph.D. thesis, National University of Singapore.

GB/T 13715–92. 1993. Contemporary Chinese language word-segmentation specification for information processing. Technical report, , Beijing.

Ge, Xianping, Wanda Pratt, and Padhraic Smyth. 1999. Discovering chinese words from unsegmented text. In *SIGIR '99*, pages 271–272.

Godfrey, J., E. Holliman, and J. McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *Proceedings of IEEE ICASSP*, volume 1, pages 517–520, Detroit.

Good, I. 1953. The population frequencies of species and the estimation of population parameters. *Biometrica V*, 40(3,4):237–264.

Greenberg, Steven and Eric Fosler-Lussier. 2000. The uninvited guest: Information's role in guiding the production of spontaneous speech. In *Proceedings of the Crest Workshop on Models of Speech Production: Motor Planning and Articulatory Modelling*, Germany.

Gu, Ping and Yuhang Mao. 1994. Hanyu zidong fenci de jinlin pipei suanfa ji qi zai QHFY hanying jiqi fanyi xitong zhong de shixian [The adjacent matching algorithm of Chinese automatic word segmentation and its implementation in the QHFY Chinese-English system. In *International Conference on Chinese Computing*, Singapore.

Harrison, Michael. 1978. *Introduction to Formal Language Theory*. Addison Wesley, Reading, MA.

Hockenmaier, Julia and Chris Brew. 1998. Error driven segmentation of chinese. *Communications of COLIPS*, 8(1):69–84.

Hopcroft, John and Jeffrey Ullman. 1979. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, Reading, MA.

Huang, C.-T. James. 1984. Phrase structure, lexical integrity and Chinese compounds. *Journal of the Chinese Language Teachers Association*, 14:53–78.

Huang, Chu-Ren. 1999. From quantitative to qualitative studies: Developments in computational and corpus linguistics of Chinese. Institute of Linguistics, Academia Sinica.

Huang, Chu-Ren, Kathleen Ahrens, and Keh-Jiann Chen. 1994. A data-driven approach to psychological reality of the mental lexicon: Two studies on chinese corpus linguistics. In *Language and its Psychobiological Bases*, Taipei.

Huang, Chu-Ren, Keh-Jiann Chen, Chang Lili, and Feng-yi Chen. 1997. Segmentation standard for Chinese natural language processing. *International Journal of Computational Linguistics and Chinese Language Processing*, 2(2):47–62.

Huang, Chu-Ren, Wei-Mei Hong, and Keh-Jiann Chen. 1994. Suoxie: An information based lexical rule of abbreviation. In *Proceedings of the Second Pacific Asia Conference on Formal and Computational Linguistics II*, pages 49–52, Japan.

Kaplan, Ronald and Martin Kay. 1994. Regular models of phonological rule systems. 20:331–378.

Kiraz, George. 1999. *Computational Approach to Non-Linear Morphology*. Cambridge University Press, Cambridge.

Klatt, D. H. 1975. Vowel lengthening is syntactically determined in a connected discourse. *Journal of Phonetics*, 3:129–140.

Kochanski, Greg P and Chilin Shih. 2001a. Hierarchical structure and word strength prediction of mandarin prosody. In *4th ISCA Tutorial and Research Workshop on Speech Synthesis*, Scotland, UK.

Kochanski, Greg P. and Chilin Shih. 2001b. Soft templates for prosody mark-up. *Submmited to Speech Communications*.

Kruskal, Joseph B. 1983. An overview of sequence comparison. In David Sankoff and Joseph Kruskal, editors, *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. chapter 1, pages 1–44.

Lehiste, I. 1972. Timing of utterances and linguistic boundaries. *JASA*, 51:2018–2024.

Lewis, Harry and Christos Papadimitriou. 1981. *Elements of the Theory of Computation*. Prentice-Hall, Englewood Cliffs, NJ.

Li, B.Y., S. Lin, C.F. Sun, and M.S. Sun. 1991. Yi zhong zhuyao shiyong yuliaoku biaoji jinxing qiyi jiaozheng de zuida pipei hanyu zidong fenci suanfa sheji [a maximum-matching word segmentation algorithm using corpus tags for disambiguation]. In *ROCLING IV*, pages 135–146, Taipei. ROCLING.

Li, Charles and Sandra Thompson. 1981. *Mandarin Chinese: A Functional Reference Grammar*. University of California Press, Berkeley, CA.

Liang, Nanyuan. 1986. Shumian hanyu zidong fenci xitong-CDWS [a written Chinese automatic segmentation system–CDWS]. *Journal of Chinese Information Processing*, 1(1):44–52.

Lin, Ming-Yu, Tung-Hui Chiang, and Keh-Yi Su. 1993. A preliminary study on unknown word problem in Chinese word segmentation. In *ROCLING 6*, pages 119–141. ROCLING.

Lisker, L. 1957. Closure duration and the intervocalic voiced-voiceless distinction in English. *Language*, 33:42–29.

Liu, Yi and Pascale Fung. 2000. Modeling pronunciation variations in spontaneous mandarin speech. In *ICSLP 2000*, Beijing, China. ICSLP.

Manning, Christopher and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.

Mitchell, Tom M. 1997. *Machine Learning*. WCB/McGraw-Hill, Boston, MA.

Mohri, Mehryar. 1994. Syntactic analysis by local grammars automata: an efficient algorithm. In *Papers in Computational Lexicography: COMPLEX '94*, pages 179–191, Budapest. Research Institute for Linguistics, Hungarian Academy of Sciences.

Mohri, Mehryar. 1997. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2).

Nie, Jian-Yun, Wanying Jin, and Marie-Louise Hannan. 1994. A hybrid approach to unknown word detection and segmentation of Chinese. In *International Conference on Chinese Computing*, Singapore.

Packard, Jerome. 2000. *The Morphology of Chinese: A Linguistics and Cognitive Approach*. Cambridge University Press, Cambridge.

Palmer, David. 1997. A trainable rule-based algorithm for word segmentation. In *Proceedings of the Association for Computational Linguistics*.

Palmer, David and John Burger. 1997. Chinese word segmentation and information retrieval. In *AAAI-97*.

Pan, Wenguo (潘文國), Po-Ching Yip (葉步青), and Yang Saxena Han (韓洋). 1993. 漢語的構詞法研究 *(Studies in Chinese Word-Formation)*. Student Book Company, Taipei.

Peng, Z.-Y. and J-S. Chang. 1993. Zhongwen cihui qiyi zhi yanjiu — duanci yu cixing biaoshi [Research on Chinese lexical ambiguity — segmentation and part-of-speech tagging]. In *ROCLING 6*, pages 173–193. ROCLING.

Öhman, S. 1967. Word and sentence intonation, a quantitative model. Technical report, Department of Speech Communication, Royal Institute of Technology (KTH).

Quinlan, J. R. 1986. Induction of decision trees. *Machine Learning*, 1(1):81–106.

Ren, Hongmo. 1985. Linguistically conditioned duration rules in a timing model for Chinese. In Ian Maddieson, editor, *UCLA Working Papers in Phonetics 62*. UCLA, pages 34–49.

Sankoff, David and Joseph Kruskal, editors. 1983. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley Publications, csli publications. the david hume series, philosophy and cognitive science reissues. 1999 edition.

Shannon, Claude E. 1948. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, July, October.

Shih, Chilin and Benjamin Ao. 1997. Duration study for the Bell Laboratorie s Mandarin text-to-speech system. In Jan van Santen, Richard Sproat, Joseph Olive, and Julia Hirschberg, editors, *Progress in Speech Synthesis*. Springer, New York.

Shih, Chilin and Richard Sproat. 1996a. Issues in text-to-speech conversion for Mandarin. *Computational Linguistics and Chinese Language Processing*, 1:37–86.

Shih, Chilin and Richard Sproat. 1996b. Issues in text-to-speech conversion for mandarin. *Computational Linguistics and Chinese Language Processing*, 1(1):37–86.

Smadja, Frank. 1993. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19(1):143–177.

Sproat, Richard. 1992. *Morphology and Computation*. MIT Press, Cambridge, MA.

Sproat, Richard. 2000. *A Computational Theory of Writing Systems*. Cambridge University Press, Stanford, CA.

Sproat, Richard, Alan Black, Stanley Chen, Shankar Kumar, Mari Ostendorf, and Christopher Richards. 2001. Normalization of non-standard words. *Computer Speech and Language*, 15(3).

Sproat, Richard and Chilin Shih. 1990. A statistical method for finding word boundaries in Chinese text. *Computer Processing of Chinese and Oriental Languages*, 4:336–351.

Sproat, Richard and Chilin Shih. 1996. A corpus-based analysis of Mandarin nominal root compounds. *Journal of East Asian Linguistics*, 4(1):1–23.

Sproat, Richard, Chilin Shih, William Gale, and Nancy Chang. 1996. A stochastic finite-state word-segmentation algorithm for Chinese. *Computational Linguistics*, 22(3).

Stevens, K. N. 1998. *Acoustic Phonetics*. The MIT Press.

Sun, Maosong, Dayang Shen, and Benjamin K. Tsou. 1998. Chinese word segmentation without using lexicon and hand-crafted training data. In *Proceedings of COLING-ACL '98*, pages 1265–1271.

Teahan, W. J., Yingying Wen, Rodger McNab, and Ian Witten. 2000. A compression-based algorithm for Chinese word segmentation. *Computational Linguistics*, 26(3):375–393.

Torgerson, Warren. 1958. *Theory and Methods of Scaling*. Wiley, New York.

van Santen, Jan P. H. 1993. Analyzing n-way tables with sums-of-products models. *Journal of Mathematical Psychology*, 37:327–371.

van Santen, Jan P. H. and Joe P. Olive. 1990. The analysis of contextual effects on segmental duration. *Computer Speech and Language*, 4:359–391.

van Santen, Jan P. H. and Chilin Shih. 2000. Suprasegmental and segmental timing models in mandarin chinese and american english. *JASA*, 107(2):1012–1026.

Wang, Kuijing (王魁京). 1996. 現代漢語縮略語詞典 *(A Dictionary of Present-Day Chinese Abbreviations)*. Shangwu Printing House, Beijing.

Wang, Liang-Jyh, Wei-Chuan Li, and Chao-Huang Chang. 1992. Recognizing unregistered names for mandarin word identification. In *Proceedings of COLING-92*, pages 1239–1243. COLING.

Wang, Yongheng, Haiju Su, and Yan Mo. 1990. Automatic processing of chinese words. *Journal of Chinese Information Processing*, 4(4):1–11.

Weeber, Marc, Rein Vos, and Harald Baayen. 2000. Extracting the lowest-frequency words: Pitfalls and possibilities. *Computational Linguistics*, 26(3):301–317.

Wu, Zimin and Gwyneth Tseng. 1993. Chinese text segmentation for text retrieval: Achievements and problems. *Journal of the American Society for Information Science*, 44(9):532–542.

Xia, Fei. 1999. Segmentation guideline, Chinese Treebank Project. Technical report, University of Pennsylvania. `http://morph.ldc.upenn.edu/ctb/`.

Xu, Yi and X. J. Sun. 2000. How fast can we really change pitch? maximum speed of pitch change revisited. In *ICSLP*.

Yeh, Ching-long and Hsi-Jian Lee. 1991. Rule-based word identification for Mandarin Chinese sentences — a unification approach. *Computer Processing of Chinese and Oriental Languages*, 5(2):97–118.

Zipf, George Kingsley. 1949. *Human Behavior and the Principle of Least Effort*. Addison-Wesley.