Computational Linguistics and Chinese Language Processing
Vol. 8, No. 1, February 2003, pp. 1-28 *1*

Customizable Segmentation of Morphologically Derived Words in Chinese

Andi Wu*

Abstract

The output of Chinese word segmentation can vary according to different linguistic definitions of words and different engineering requirements, and no single standard can satisfy all linguists and all computer applications. Most of the disagreements in language processing come from the segmentation of morphologically derived words (MDWs). This paper presents a system that can be conveniently customized to meet various user-defined standards in the segmentation of MDWs. In this system, all MDWs contain word trees where the root nodes correspond to maximal words and leaf nodes to minimal words. Each non-terminal node in the tree is associated with a resolution parameter which determines whether its daughters are to be displayed as a single word or separate words. Different outputs of segmentation can then be obtained from the different cuts of the tree, which are specified by the user through the different value combinations of those resolution parameters. We thus have a single system that can be customized to meet different segmentation specifications.

Keywords: segmentation standards, morphologically derive words, customizable systems, word-internal structures

1. Introduction

A written sentence in Chinese consists of a string of evenly spaced characters with no delimiters between the words[1]. In any word-based Chinese language processing[2], therefore, segmenting each sentence into words is a prerequisite. However, due to some special linguistic properties of Chinese words, there is not a generally accepted standard that can be

* Microsoft Research
Address: 21062 NE 81st Street, Redmond, WA, USA
E-mail: andiwu@microsoft.com
Phone: (O) 1-425-706-0985 (H) 1-425-868-8075

[1] See Sproat [2000] for a theoretical account of this orthographic convention.

[2] Character-based processing is also possible and has performed well in certain applications.

used to unambiguously determine "wordhood" in every case.[3] While native speakers of Chinese are often able to agree on how to segment a string of characters into words, there are a substantial number of cases where no agreement can be reached [Sproat *et al*. 1996]. Besides, different natural language processing (NLP) applications may have different requirements that call for different definitions of words and different granularities of word segmentation. This presents a challenging problem for the development of annotated Chinese corpora that are expected to be useful for training multiple types of NLP systems. It is also a challenge to any Chinese word segmentation system that claims to be capable of supporting multiple user applications. In what follows, we will discuss this problem mainly from the viewpoint of NLP and propose a solution that we have implemented and evaluated in an existing Chinese NLP system[4].

In Section 2, we will look at the problem areas where disagreements among different standards are most likely to arise. We will identify the alternatives in each case, discuss the computational motivation behind each segmentation option, and suggest possible solutions. This section can be skipped by readers who are already familiar with Chinese morphology and the associated segmentation problems. Section 3 presents a customizable system where most of the solutions suggested in Section 2 are implemented. The implementation will be described in detail and evaluation results will be presented. We also offer a proposal for the development of linguistic resources that can be customized for different purposes. In Section 4, we conclude that, with the preservation of word-internal structures and a set of resolution parameters, we can have a Chinese system or a single annotated corpus that can be conveniently customized to meet different word segmentation requirements.

2. Target Areas for Customization

How to identify words in Chinese has been a long-standing research topic in Chinese linguistics and Chinese language processing. Many different criteria have been proposed and any serious discussion of this issue will take no less than a book such as [Packard 2000]. Among the reasons that make this a hard and intriguing problem are:

- Chinese orthography has no indication of word boundaries except punctuation marks.
- The criteria for wordhood can vary depending on whether we are talking about the phonological word, lexical word, morphological word, syntactic word, semantic word, or psychological word [Packard 2000, Di Sciullo and Williams 1987, Dai 1992, Dai 1997, Duanmu 1997, Anderson 1992, Sadock 1991, Selkirk 1982, *etc*.].

---

[3] For a comprehensive review of this problem, see Packard [2000].

[4] This system is developed at Microsoft Research in the general framework of Jensen *et al* [1993] and Heidorn [2000]. Details of the Chinese system can be found in Wu *et al* [2000, 1998].

- Unlike Japanese, Chinese has very little inflectional morphology that can provide clues to word boundaries.
- Many bound morphemes in Chinese used to be free morphemes and they are still used as free morphemes occasionally. Therefore the distinction between bound morphemes and words can be fuzzy.
- The character sequence of many Chinese words can be made discontinuous through morphological processes.
- Word-internal structures look similar to syntactic structures. As a result, there is often confusion between words and phrases [Dai 1992].
- Structural information is not always sufficient for identifying a sequence of characters as a word. Frequency of the sequence, mutual information between the component syllables, and the number of syllables in that sequence also play a role (Summarized in [Sproat 2002]).

As a result, native speakers of Chinese often disagree on whether a given character string is a word. As reported in [Sproat *et al*, 1996], the rate of agreement among human judges was only 76%. It is not hard to understand, then, why Chinese linguists have had such a hard time defining words.

However, we do not have to wait for linguists to reach a consensus before we do segmentation in NLP. In computer applications, we are more concerned with "segmentation units" than "words". While words are supposed to be well-defined, unambiguous and static linguistic entities, segmentation units are not. In fact, segmentation units are expected to vary from application to application. In information retrieval, for example, the segmentation units are search terms, whose sizes may vary according to specific needs. A system aimed at precision will require "larger" units while a system aimed at recall will require "smaller" ones. A good Chinese IR system should be flexible with the output of word segmentation so that search terms of different sizes can be generated. In machine translation, the segmentation units are strings that can be mapped onto the words of another language. An MT system should not be committed to a single segmentation, since the granularity of that segmentation may be good for some mappings but not for others. We can do better if a variety of segmentation units are generated so that all possible words are made available as candidates for alignment. In an N-gram language model, the segmentation units are the "grams" and their sizes may need to be adjusted against the perplexity of the model or the sparseness of data. In text-to-speech systems, the segmentation units can be prosodic units and the units that are good for IR may not be good for TTS. In short, a segmentation system can be much more useful if it can provide alternative segmentation units. Alternative units provide linguistic information at different levels and each alternative can serve a specific purpose. We

will see some concrete examples in the remainder of this section. To facilitate the use of terminology, we will use "words" to mean "segmentation units" in the rest of this paper.

Now where does the variability in segmentation units come from? If we compare the outputs of various word segmentation systems, we will find that they actually have far more similarities than differences. This is mainly due to the fact that the word lists used by different segmenters have a lot in common. The actual differences we observe usually involve words that are not typically listed in the dictionary. These words are more dynamic in nature and are usually formed through productive morphological processes. It is those morphologically derived words (MDWs hereafter) that are most controversial and most likely to be treated differently in different standards and different systems. This is the main focus of this paper.

The morphological processes we will be looking at have all been discussed extensively in the literature and a brief summary of them can be found in [Sproat 2002]. We will not attempt to review the literature here. Instead, we will concentrate on cases where differences in segmentation are likely to arise. Here are the main categories of morphological processes we will go through:

- Reduplication
- Affixation
- Directional and resultative compounding
- Merging and splitting
- Named entities and factoids

During the discussion, we will make frequent reference to the following four existing segmentation standards:

(1) The segmentation guidelines for the Penn Chinese Treebank [Xia 2000] ("CHTB" hereafter).

(2) The guidelines for the Beijing University Institute of Computational Linguistics Corpus [Yu 1999] ("BU" hereafter). These guidelines closely follow the GB standard [GB/T 13715-92, 1993] but have some additional specifications.

(3) The ROCLING standard developed at Academia Sinica in Taiwan. [Huang *et al*. 1997, ROCLING 1997] ("ROCLING" hereafter).

(4) The standard used in our own system.

Our segmentation system is developed as an integral part of a Chinese parser where initial word segmentation produces a weighted word lattice. The word lattice contains all the dictionary words plus the MDWs formed by morphological rules. Syntactic parsing takes this word lattice as its input and the final segmentation corresponds to the leaves of the best parse

tree[5]. Segmentation ambiguities are resolved in the parsing process and the correct segmentation is the one that enables a successful parse. In cases where parsing fails, we back off to partial parsing and use dynamic programming to assemble a tree that consists of the largest partial trees.

2.1 Reduplication

The main patterns of reduplication in Chinese are AA, ABAB, AABB, AXA, AXAY, XAYA, AAB and ABB. Examples of these patterns can be found in Appendix 1. Existing standards do not have much disagreement over the segmentation of AA, AABB, AXAY, XAYA, AAB and ABB. These are all considered single words for the simple reason that, except in the case of AA, breaking them up will result in segments that are not independent words. The problem cases are ABAB and AXA.

2.1.1 ABAB

A representative example of this is "讨论讨论" (taolun-taolun: discuss-discuss "*have a discussion*"). It is considered a single word in the CHTB and ROCLING standards, but two separate words in the BU standard. According to CHTB and ROCLING, ABAB is just a variation of AA, where the reduplicated word is made of two characters instead of one. Since the meaning of AA (such as "看看" (kan-kan: look-look "*take a look*")) or ABAB is not compositional,[6] they should be both considered single words. According to the BU standard, however, "讨论讨论" should be broken up because "讨论" can be looked up in the dictionary but "讨论讨论" can not.

Different NLP applications can also have different requirements. The one-word segmentation may simplify syntactic analysis but the two-word segmentation might be better for information retrieval or word-based statistical summarization. For pinyin-to-character conversion, adding the reduplicated form to the word list should improve accuracy but may not have the desired effect if the data is too sparse. In machine translation, it will be desirable to have both: the one-word analysis will make it easier for us to learn mappings between, say, "讨论讨论" and "have a discussion", whereas the two-word analysis will let us translate "讨论" into "discuss" in case no mapping is found for "讨论讨论" in the training data. In our system, we treat ABAB as a single word with internal structure, i.e. [讨论 讨论], so that we can have access to both kinds of information. The word also has a "lemma" attribute indicating that the "underlying form" is "讨论".

---

[5] The weights in the word lattice are considered in the selection of the best parse.

[6] The meaning of AA is not "A and A". The verb or adjective is duplicated here to represent certain grammatical aspects, such as short duration or attempted action.

2.1.2 AXA

This covers cases like the following:

试一试	shi-yi-shi: try-one-try	"*give it a try*"
试了试	shi-le-shi: try-LE-try[7]	"*gave it a try*"
试了一试	shi-le-yi-shi: try-LE-one-try	"*gave it a try*"

Both BU and ROCLING regard those expressions as separate words, while CHTB treats them as single words with internal structures. Our system also analyzes them as single words. To represent the fact that AXA is an instance of A with additional aspectual information, we store two additional attributes in this word: a "lemma" attribute that holds the "underlying form" of the MDW (e.g. "试" for "试了试") and an "aspect" attribute whose value(s) record the aspectual information carried by "一" and/or "了".

The lemma attribute is in fact assigned in each type of reduplication. This is especially important for AABB, AAB and ABB. In the case of AABB such as "清清楚楚" (qing-qing-chu-chu "*very clear*"), for instance, we will not get "清楚" (qingchu "clear") unless we segment it into "清 / 清楚 / 楚" which is not acceptable by any standard because of the dangling bound morphemes on the two sides. This problem disappears once we have "清楚" represented as the lemma of the whole reduplicated form.

2.2 Affixation

Affixation is a very productive morphological process in Chinese. Examples of various derivational processes can be found in Appendix II. As we can see, the morphological rules that combine stems with affixes are almost indistinguishable from the syntactic rules that attach a modifier to a head. The only difference is that the modifier (in the case of prefixation) or the head (in the case of suffixation) is supposed to be a bound morpheme. However, the line between free morphemes and bound morphemes is often hard to draw in Chinese.[8] There are some relatively clear cases, such as 非 (fei "*non-*") and 超 (chao "*super-*") as prefixes and 者 (zhe "*-er*") and 学 (xue *"-ology"*) as suffixes, but the distinction is fuzzy in many cases.

---

[7] Function words like 了 have no English translation and therefore will be glossed by the uppercase versions of their pronunciation.

[8] Here are a few borderline cases:

总工程师	zong-gongchenshi	"*chief engineer*"
副主席	fu-zhuxi	"*vice-chairman*"
足球场	zuqiu-chang	"*soccer field*"
警察局	jingcha-ju	"*police station*"
煤气炉	meiqi-lu	"*gas stove*"

Are they words or phrases?

Even the agentive suffix 者 can act as a free morpheme in cases like "持枪闯入民宅者" (chi-qiang-chuang-ru-min-zhai-zhe: carry-gun-break-into-civilian-residence-er "*people who broke into houses with guns*") where 者 is the head of a noun phrase modified by a relative clause. To avoid this thorny issue, different segmentation standards resorted to different definitions of affixation. In the CHTB standard, the term "affixation" is not explicitly used. Instead, it describes prefixation as JJ+N where JJ is monosyllabic, and suffixation as N+N where the second N is monosyllabic. The ROCLING standard distinguishes between affixes, "word beginning" (接头词 jietouci) and "word endings" (接尾词 jieweici), but they are functionally equivalent in derivational rules. The BU standard tries to distinguish between affixation and modifier-head phrases by restricting affixation to words that end in a pre-specified list of affixes.

In terms of segmentation, all the standards agree that MDWs derived from affixation should be treated as single words. In actual NLP applications, however, we often wish to have access to both the derived word as a whole as well as its components as separate words. In machine translation, for instance, it might be desirable to have a choice of translating either the whole or the parts: translate the whole if a translation for the whole can be found and back off to the parts otherwise. Take 烘干机 (honggan-ji: dry-machine "*dryer*") as an example. Ideally the whole word should be translated into "dryer". However, if our translation knowledge base has no translation for 烘干机 but does have translations for 烘干 and 机, we should be able to translate it as "drying machine" given that the parts are also available. In information retrieval, we may also want to search for the parts if the query term as a whole is not found. For example, we may want to retrieve texts containing 警察 (jingcha "*police*") when the query term is 警察局 (jingcha-ju:police-bureau, "*police station*") .

In our system, we treat complex words derived from affixation as single words, just as the other standards do, but we also keep their internal structures. For example, the complex word 核物理学家 (he-wuli-xue-jia: nuclear-physics-science-expert "*nuclear- physicist*") is represented as [[[核 物理] 学] 家]. Each derived word contains such as a sub-tree. The sub-tree functions as a single leaf node in syntactic analysis but it can be made visible after parsing to become part of the parse tree if necessary.

2.3 Directional and Resultative Compounding

There are many kinds of compounding in Chinese. In terms of word segmentation, the most problematic ones are directional compounding and resultative compounding. In directional compounding, a verb is followed by a directional complement, such as 上 (shang, "*up*"), 下 (xia "*down*"), 进去 (jinqu "*into*"), 出来 (chulai "*out*"), which indicates the direction of the action expressed by the verb. In resultative compounding, a verb is followed by a resultative complement which is a verb or adjective that indicates what results from the action of the first

verb. In both cases, the verb and the complement can be separated by 得 (de) or 不 (bu) to express the possibility of the verb-complement relationship. Here are some examples:

Directional compounding:

走进	zou-jin: walk-enter	"*walk into*"
走进去	zou-jinqu: walk-enter	"*walk in*"
走得进去	zou-de-jinqu: walk-DE-enter	"*can walk in*"

Resultative compounding:

带走	dai-zou : take-go	"*take away*"
带得走	dai-de-zou: take-DE-go	"*can take away*"
带不走	dai-bu-zou: take-not-go	"*cannot take away*"
看清楚	kan-qingchu: see-clear	"*see clearly*"
看得清楚	kan-de-qingchu: see-DE-clear	"*can see clearly*"
看不清楚	kan-bu-qingchu: see-not-clear	"*cannot see clearly*"

The segmentation of those compounds depends on many factors:

(1) Type of compounding. Directional compounds are more likely to be treated as single words than resultative compounds. Both CHTB and ROCLING follow this principle.

(2) Word length. Those compounds are more likely to be treated as separate units if their total length is more than 2. CHTB provides internal structures when the compound is longer than 2 characters. ROCLING treats "看清" (kan-qing: see-clear "*see clearly*") as one word but "看清楚" (kan-qingcu: see-clear "*see clearly*") as two words.

(3) Frequency. Compounds that are more frequent, either synchronically or diachronically, tend to be treated as one word. Compare 打破 (da-po: hit-break "*hit and make it break*") and 打痛 (da-tong: hit-hurt "*hit and make someone hurt*"). These two compounds have exactly the same internal structure and the same word length, but former is more likely to be regarded as a single word than the latter, simply because 打破 is more frequent. The BU standard assumes that all the frequent compounds are already in its lexicon. Therefore non-lexicalized compounds are to be broken up into independent words.

(4) Mutual information [Sproat and Shih 1990]. Compounds whose components have strong mutual information between them are usually taken as single words. For example, 撕裂 (si-lie: tear-split "*tear open*") is not as frequent as 撕坏 (si-huai: tear-bad "*tear and break*"), but 撕裂 is lexicalized in the BU dictionary while 撕坏 is not.

(5) Some resultative verbs are more independent and therefore more likely to stand on their own. Typical examples are 完 (wan "*finish*") and "给" (gei "*give*") which have some special grammatical functions[9] in addition to being resultative complements.

(6) "V + 得/不 + complement" structures are segmented into separate words in BU and ROCLING but kept as single items with internal structures in CHTB[10]. The main reason for keeping them together is that the verb and the complement can usually form a single word.

NLP applications have considerations that are not always compatible with human judgment. In machine translation, it often makes more sense to break up directional compounds into independent words and keep resultative compounds as single words, contrary to the tendencies we observed above. Directional compounds often correspond to verb-preposition sequences in other languages. The compound "走进", for example, corresponds to "walk into" in English. If "走进" is segmented into two words, we will be able to align "走" with "walk" and "进" with "into". After seeing other instances of Verb+进, such as "跑进" (pao-jin: run-enter "*run into*") and "跳进" (tiao-jin: jump-enter "jump into"), we can come to the generalization that Verb+进 is to be translated as Verb+into in English. If those compounds are reduced to single words, we can still learn the correspondence between "走进" and "walk into", but the generalization is not so easy to reach. Resultative compounds, on the other hand, are much more likely to correspond to single words in languages that are unrelated to Chinese. "打破", for example, will most likely align with "break" in English rather than "hit and break" or "break by hitting".

In the case of "V + 得/不 + complement" structures, it is important to know the relationship between the verb and the complement. We need a representation where 吃得下 (chi-de-xia: eat-DE-down "can eat up"), for instance, can be interpreted as having more or less the same meaning of "能 吃下" (neng-chi-xia: can-eat-down "can eat up"). This is crucial not only for semantic analysis, but for such seemingly simple computer applications as various types of Chinese input methods where a language model is used to select the best sequence of characters. Most existing IME systems are error-prone when the input contains the "V + 得/不 + complement" structure. They are unable to relate the verb and the complement even though the verb-complement bigram is in the language model.

To meet the needs of as many standards and applications as possible, our system treats all directional and resultative compounds as single words while preserving their internal

---

[9] 完 can be viewed as an aspectual marker indicating the completion of an action while 给 may have a role similar to the English "to" in dative constructions.

[10] Except in cases like 吃不了 (chi-bu-liao:eat-not-done "*unable to eat anymore*") where V+complement" is not a legitimate compound.

structures. In cases of "V + 得/不 + complement", we also represent the "lemma" which is equivalent to "V + complement". The result is a word tree, where the root node contains the lemma of the compound.

2.4 Merging and Splitting

Both merging and splitting result in word fragments, which often creates a dilemma as to whether to keep those strings as single units or not. We will look at them one by one.

2.4.1 Merging

This morphological process, also known as "telescopic compounding" [Huang *et al*. 1997], can be considered a sub-case of abbreviation, but unlike other kinds of abbreviation, it has a fixed pattern and a predictable semantic interpretation. It applies to cases where two adjacent and semantically related words have some characters in common. The common characters may be at the beginning or end of the words. Here are some examples.

Common beginnings (AB+AC => ABC)

国内+国外 => 国内外 guo-nei-wai: country-inside-outside

"domestic + foreign" => "domestic and foreign"

Common endings (AC+BC => ABC)

进口+出口 => 进出口 jin-chu-kou: enter-exit-port

"import + export" => "import and export"

Ending = Beginning (AB+BC => ABC)

上海市+市长 => 上海市长 shanghai-shi-zhang: Shanghai-city-head

"Shanghai City + city mayor" => "mayor of Shanghai"

All existing standards agree that we have a single word in the AB+AC and AC+BC cases[11] and two words in the AB+BC case. The problem in the first two cases is that, unless we store ABC in the dictionary as a whole, we will not be able to assign good semantic interpretations to them. However, not all words of this kind can be stored in the dictionary, since merging is a productive morphological process. To interpret a newly merged word, such as 存贷款 (cun-dai-kuan: deposit-borrow-fund "*deposits and loans*"), which is unlikely to be in the dictionary, we seem to need a level of representation where ABC shows up in its underlying form, i.e. AB AC or AC BC. 存贷款 should then be represented as 存款 贷款, not at the surface segmentation, but as the "lemmas" of 存贷款. This is what we do in our system where every merged word contains a tree where the lemmas are conjoined.

[11] Unless the sequence is interrupted by a punctuation mark, as in 国内、外 and 进、出口.

2.4.2 Splitting

Splitting is an active morphological process where a multiple-character word with an internal verb-object structure is split into two non-consecutive parts by the insertion of an aspect marker, a measure word or other functional elements. Here are some examples:

Insertion of an aspect marker

洗了澡 xi-le-zao: wash-LE-bath "*took a bath*"

Insertion of a measure word

洗个澡 xi-ge-zao: wash-one-bath "*take a bath*"

Insertion of both an aspect marker and a measure word

洗了个澡 xi-le-ge-zao: wash-LE-one-bath "*took a bath*"

Insertion of even more words

洗了个舒舒服服的澡 xi-le-ge-shushufufu-de-zao: wash-LE-one-comfortable-DE-bath "*took a comfortable bath*"

Most segmentation standards require such expressions to be segmented into multiple words, such as 洗 / 了 / 澡. This can result in segments that are not independent words, as we see in the case of 澡 which is a bound morpheme. One may argue that in such cases the bound morpheme is acting as a free morpheme. But it would still be desirable to have a representation which indicates that 洗 and 澡 actually form a single word and 洗了澡 has more or less the same meaning as 洗澡+了. In other words, the lemma of 洗了澡 should be 洗澡. Such a representation can be difficult in the case of 洗了个舒舒服服的澡, but even there 洗 and 澡 still form a single unit in some sense.

The lemma representation of a split word is obviously useful in the realm of information retrieval since it makes it possible to establish links between the split and non-split forms of the same verbs. As in the verb-complement case (2.3), it may also be beneficial to Chinese input methods that use an N-gram language model to select the correct character sequences. Most existing systems perform poorly when the input contains split words. While the non-split forms of those words (such as 洗澡) are usually in the N-gram model, the split forms are not. If future systems employ word segmentation where the split form is recognized as a single unit with its lemma represented, we will be able to relate 洗 and 澡 in 洗了澡 as long as we have the bigram "洗澡" in the model.

A special case of splitting is found in expressions like 跳起舞来 (tiao-qi-wu-lai "*start dancing*") where two words (跳舞 and 起来 in this case) cross each other. Here again we need a level of representation to encode the fact that 跳起舞来 actually means 跳舞+起来.

Our system regards a split word as a single unit with a single lemma and a subtree if the intervening characters are no more than 2. Syntactic analysis treats the unit as a single leaf

and has the option of exposing the subtree as part of the parse tree after parsing is done. For cases like 洗了个舒舒服服的澡, we parse them as separate words and, if 澡 is found to be the object of 洗 in the parse, we will concatenate the lemmas of the verb and the object (i.e. 洗+澡), look up 洗澡 in the dictionary, and make it the lemma of the subtree if it exists as a dictionary entry. This can also be done in the case of 洗了澡 but we choose to make it a single unit at the lexical level just to reduce the complexity of syntactic analysis. Once its subtree (which also has the verb-object structure in it) is merged into the main parse, we will have a unified representation for 洗了澡 and 洗了个舒舒服服的澡.

2.5 Named entities and factoids

This is an area with the greatest amount of variation among segmentation standards. This is also an area where linguistic theory has very little to say on the justification of a given standard. The differences are mostly computationally motivated and the main concern here is the granularity of segmentation. Different segmentation standards prefer different levels of granularity, but the differences are fairly systematic and can be easily specified in segmentation guidelines. Listed below are the most common types of named entities and factoids whose segmentation may vary across different standards.

2.5.1 Personal names

A personal name is usually composed of a first name and a last name. The BU standard segments a Chinese name into these two parts and treats a foreign name as a single unit if the first name and last name are connected by "· ", as in "诺罗敦· 西哈努克 nuoluodun-xihanuke "*Norodom Sihanouk*". Other standards treat both Chinese and foreign names as single words. In our system, a personal name is a single word with an internal structure which indicates not only the family name and the given name but the components of the given name as well.

2.5.2 Place names and organization names

There are many levels of granularity here. For instance, "江苏省盐城地区" (jiangsu-sheng-yancheng-diqu: Jiangsu-province-Yancheng-prefecture, "*Yancheng Prefecture, Jiangsu Province*") can be segmented as "江苏省盐城地区", "江苏省 / 盐城地区", "江苏省 / 盐城 / 地区" or "江苏 / 省 / 盐城 / 地区". Likewise, "世界贸易组织" (shijie-maoyi-zuzhi: world-trade-organization "World Trade Organization") can be segmented as "世界贸易组织" or "世界 / 贸易 / 组织". Existing standards usually break those names up as long as it does not result in single-character segments. So place names with single-character place-type suffixes (such as 江苏省) tend to be kept as one word while place names with multiple-character place-type suffixes (such as 盐城地区) will be separate words. The BU standard has additional annotation to represent the internal structure of place names. "世界贸易组织", for

example, is tagged as [世界/n 贸易/n 组织/n]nt.

Each level of granularity has its pros and cons. On the one hand, "世界贸易组织" has a better chance of being aligned with "WTO" in the automatic acquisition of translation knowledge if it is segmented as one word. On the other hand, "江苏省" can be more easily related to "江苏" in information retrieval or automatic summarization if it is segmented into two words. All of this points to the need of a hierarchical structure for all the place names and organization names that contain multiple words. This is what has been done in our system.

2.5.3 Factoids

Word trees are also needed for numbers and other factoids. The reasons are obvious and therefore we will simply list some common cases where internal structures exist and different kinds of segmentation are possible.

- Numbers

  四百五十 si-bai-wu-shi-liu: four-hundred-five-ten-six " *four hundred and fifty-six*"

  四百五十六; 四百 / 五十六; 四百 / 五十 / 六; 四 / 百 / 五 / 十 / 六

  三分之一 san-fen-zhi-yi: three-divide-ZHI-one "*one third*"

  三分之一; 三 / 分之 / 一; 三 / 分 / 之 / 一

  三十多 san-shi-duo: three-ten-more "*thirty or so*"

  三十多; 三十 / 多; 三 / 十 / 多;

  数千 shu-qian:several-thousand "*several thousand*"

  数千; 数 / 千

- Dates

  一九九七年三月五日 yijiujiuqi-nian-san-yue-wu-ri: 1997-year-3-month-5-date

  "March 5, 1997"

  一九九七年三月五日; 一九九七年 / 三月 / 五日; 一九九七 / 年 / 三 / 月 / 五 / 日;

- Time

  十点零五 shi-dian-ling-wu-fen: ten-clock-zero-five-minute "*five minutes past ten*"

  十点零五分; 十点 / 零 / 五分; 十 / 点 / 零 / 五 / 分;

- Money

  六块九毛三 liu-kuai-jiu-mao-san: six-dollar-nine-dime-three

  "Six dollars and ninety-three cents"

  六块九毛三; 六块 / 九毛 / 三; 六 / 块 / 九 / 毛 / 三

- Scores

  三比一 san-bi-yi: three-match-one "*three to one*"

  三比一; 三 / 比 / 一

- Range

  三至五天 san-zhi-wu-tian: three-to-five-day "*three to five days*"

  三至五 / 天; 三 / 至 / 五 / 天; 三 / 至 / 五天

These are just simple cases. The structure can be much more complicated when one kind of named entity is embedded in another. However, no matter how complicated they are, clear guidelines can be set up to make them segmented consistently as long as their internal structures are available.

3. A Customizable System

In this section, we give a detailed description of how our system has been designed to address the problems and requirements discussed in the previous section. We will see how the word-internal structures are built, how the system can be customized to produce different outputs, and what the initial evaluation results are. Suggestions will also be made as to how the design principle here can be applied to the development of annotated corpora.

3.1 Dynamic Words

There are two types of words in our system: static words and dynamic words. Generally speaking, static words are those words that are stored in the dictionary while dynamic words are constructed at run time. All the MDWs belong in the category of dynamic words. These words are not supposed to be stored as headwords in our lexicon. Instead, they are to be built dynamically during sentence analysis through the application of a set of word-formation rules.

There are about 50 word-formation rules in our system, covering all the cases listed in Section 2 and more[12]. They are augmented phrase structure rules that have the form of A(conditions)+B(conditions) => C{actions} and each rule has a unique name that describes the particular morphological process involved. The rules are executed like a small grammar in a morphological parser before sentence-level parsing begins. They interact with each other, with some rules feeding into others, but they do not interact with the grammar rules used in sentence analysis.[13] The derivational history from the rule application then forms a tree that represents the internal structure of a given word. Figure 1 is the word tree for a fictional

[12] Some of these rules assemble unknown words that are not discussed in Section 2.

[13] We do have the option to run these rules together with the grammar rules, but that has been found to affect the system negatively both in efficiency and accuracy.

organization name, where the labels of non-terminal nodes represent the rules that are applied in constructing the tree.

赵元任语言学基金会

zhao-yuanren-yuyan-xue-jijin-hui:Zhao-Yuanren-language-science-fund-committee

"*Yuan-Ren Chao Linguistics Foundation*"

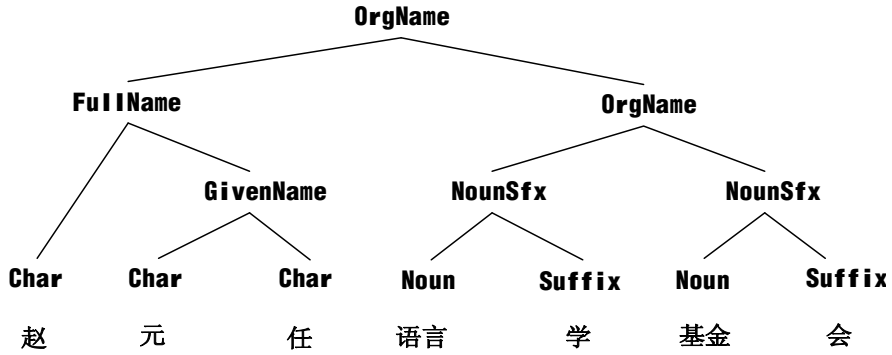

Figure 1

Trees of this kind are built for all types of MDWs, so that all of them can be treated as single words if necessary. These "maximal word trees" or "maximal words" are submitted to the sentence parser as single lexical units, which significantly reduces parsing complexity.

In all cases of merging, splitting and reduplication, the feature structure of the parent node also has an attribute that holds the lemma of the word, as we have already mentioned in Section 2. The value of the lemma is computed by piecing together the relevant characters to hypothesize a word and then checking this word against the dictionary. In the case of AABB reduplication, for instance, the hypothesized word will be AB, such as 清楚 in 清清楚楚. Since 清楚 is a word in the dictionary, it becomes the value of the lemma attribute of 清清楚楚. Similarly, the lemma of 洗了澡 is 洗澡.[14] In the case of AC+BC => ABC merging, both AC and BC will be hypothesized and put into the lemma attribute of ABC if verified in the dictionary. For example, the lemma of 进出口 is 进口+出口. These operations all take place in the "actions" part on the right-hand side of the rule.

An interesting question that arises naturally at this point is what words should be listed

[14] In addition to the lemma, we also have attributes that record the information associated with the inserted part. In 洗了澡, we store the tense/aspect information contributed by 了, so that 洗了澡 as a single verb will be equivalent to 洗了澡 as a verb phrase in terms of semantic content.

in the dictionary. According to our design, none of the MDWs should go into the dictionary. This way the word trees we get will have the maximal word at the top node, the minimal words at the leaves, and the intermediate words at the other nodes. We can thus accommodate the widest range of segmentation variations. In practice, however, there are some complications that need to be dealt with.

First of all, none of the existing dictionaries has been built strictly in line with this "minimal word" principle. They do have the minimal words, but they usually also contain words that are supposed to be dynamic in our system. It is not hard to imagine that a dictionary may contain words like 意大利式 (yidali-shi:Italy-style "*Italian-style*", 搬进 (juan-jin:move-enter "*move into*"), and 中小学 (zhong-xiao-xue:middle-small-school "*middle school and elementary school*"). Since our original dictionary was acquired rather than created in house, we do have this problem. We do not ***add*** any MDW to our dictionary, but we have to find a way to deal with those words that are already in the lexicon.

The easy way out is to leave the existing dictionary alone, with the assumption that words like 意大利式, 走进, and 进出口 are lexicalized in the dictionary because they have been lexicalized in a Chinese speaker's mind. We can also assume that they are all high-frequency words or words with strong mutual information between their components. Therefore they should stay unsegmented for probabilistic reasons. Yet another assumption is that the dictionary has listed all the exceptional MDWs that should never be segmented. If any of these assumptions turns out to be true, we should respect the dictionary entries, regarding every word in the dictionary as a minimal word, and build word trees only for words that are not in the dictionary.

These assumptions do not always hold, of course. We do find many dictionary words that can be further segmented. The solution we adopted is to keep those MDWs in the dictionary while assigning internal structures to them at run time. For all the lexicalized words that need internal structures, we mark them with two simple attributes: Type and Segs.[15] The value of Type is the name of the rule that would have been used to construct the word dynamically had this word not been lexicalized. Segs marks the potential internal word boundaries in the word. For 语言学 (yuyan-xue, language-study, "linguistics"), for example, we will have Type = "NounSfx" and Segs = "语言_学". With these two pieces of information, we are able to reconstruct the internal word tree at run time. In terms of structure, therefore, a lexicalized 语言学 will be identical to a dynamically constructed 语言学. This enables us to handle all MDWs in a unified way in later stages of processing, regardless of whether they are

[15] The addition of such information to the dictionary was done semi-automatically. We automatically extracted from the dictionary candidates for a given type of MDWs and then had a human evaluator remove the invalid ones.

from the lexicon or from the rules.

3.2 Multi-resolution parameters

Once every MDW is assigned a word tree representing its internal structure, how to segment those words becomes merely a display problem, since different segmentations of the same word can now be obtained by taking different cuts of the word tree. Borrowing a term from the graphical world, we can say that we just have to decide on the degree of "resolution" in displaying the internal structure or the granularity of output.

To control the resolution, we let every non-terminal node in the tree be associated with a multi-resolution parameter. Since every non-terminal node corresponds to a word formation rule with which the node was built, the parameter is in effect associated with a given rule or a particular type of morphological process. In the current system, those parameters are binary-valued: 0 if the daughters of a node are to be displayed as a single word and 1 if they are to be displayed as separate words. To illustrate this, we go back to the MDW in Figure 1: 赵元任语言学基金会. We find four different types of node labels in its word tree – OrgName, NounSfx, FullName and GivenName – which are the names of the rules that are used to construct this MDW. Each of them has a multi-resolution parameter: P(OrgName), P(NounSfx), P(FullName) and P(GivenName). Different settings of those parameters then result in different granularities of segmentation:

- P(OrgName) = 0:
  赵元任语言学基金会
- P(OrgName) = 1; P(NounSfx) = 0; P(FullName) = 0:
  赵元任 / 语言学 / 基金会
- P(OrgName) = 1; P(NounSfx) = 1; P(FullName) = 0:
  赵元任 / 语言 / 学 / 基金 / 会
- P(OrgName) = 1; P(NounSfx) = 0; P(FullName) = 1; P(GivenName) = 0:
  赵 / 元任 / 语言学 / 基金会
- P(OrgName) = 1; P(NounSfx) = 0; P(FullName) = 1; P(GivenName) = 1:
  赵 / 元 / 任 / 语言学 / 基金会
- P(OrgName) = 1; P(NounSfx) = 1; P(FullName) = 1; P(GivenName) = 0:
  赵 / 元任 / 语言 / 学 / 基金 / 会
- P(OrgName) = 1; P(NounSfx) = 1; P(FullName) = 1; P(GivenName) = 1:
  赵 / 元 / 任 / 语言 / 学 / 基金 / 会

We notice that the values of these parameters are not independent in a given structure.

When the parameter of a node is set to 0, the parameter values of all the nodes dominated by that node must be 0 as well. It is impossible to keep a MDW as a single word while separating some of its sub-words at the same time. The value of a parameter can be 1 only if the parameter of its parent node is set to 1. Therefore, although we have about 50 rules and consequently about 50 parameters, there do not exist 2^{50} different ways of segmenting sentences even theoretically. But we do provide enough options to adapt the segmentation to any reasonable standard. A user of our system can set those parameters according to any specification to produce the desired segmentation without making any modification in the system itself. The system is thus easily customizable.

Our current system also provides a parameter whose value determines whether word length is to be taken into consideration. As we have seen in Sections 2.3, words formed through directional and resultative compounding are sensitive to word length when it comes to segmentation. These MDWs are more likely to be treated as single words if it has fewer than three characters. The additional parameter covers this case. When it is set to 1, all MDWs built through derivational and resultative compounding will be segmented into separate words if it contains more than two characters, regardless of the values of other parameters. Suppose the name of the directional compounding rule is "DirCmpd". When the length parameter is set to 0, 走进 and 走进来 will both be kept as single words if P(DirCmpd) is set to 0. They will be segmented into two words if P(DirCmpd) is set to 1. When the length parameter is set to 1, however, 走进 will be kept as one word but 走进来 will be cut into two words even if P(DirCmpd) is set to 0.

We also added a parameter whose value determines whether the lemma or the surface string of a MDW is to be displayed. When this parameter is set to 1, the lemma will be displayed and 跳起舞来 will be displayed as 跳舞 起来. This is of course more like stemming than word segmentation, but this is a functionality that some applications may require. In fact, this might be one of the steps we have to take to go from the "truthful" level of segmentation to the "graceful" level [Huang *et al*. 1997].

3.3 Evaluation

To find out the degree of customization that can be achieved by the parameterization described above, we evaluated our system against two annotated corpora that were made publicly available for SIGHAN's First International Chinese Word Segmentation Bakeoff: the training data of the Penn Chinese Tree Bank and the Beijing University Institute of Computational Linguistics Corpus. These two annotated corpora follow very different guidelines and it should be interesting to see how well our system can adapt to them. The evaluation metric we used to measure our performance was the scoring tool written by Richard Sproat for the First International Chinese Word Segmentation Bakeoff. This scoring

tool measures word recall, word precision, the F-measure, the OOV rate, and the OOV recall rate, among other things. Given a reference (the gold standard) and a hypothesis (the segmentation hypothesized by the word segmenter), word recall is the percentage of words in the reference that are also in the hypothesis, and word precision is the percentage of words in the hypothesis that are also in the reference. The F-measure is a simple average of precision and recall. The OOV rate is the percentage of words in the reference that are not found in the dictionary, and the OOV recall rate is the percentage of OOV words that are found in the hypothesis. The OOV scores are of interest in this paper because many of the OOV words are MDWs according to our dictionary and the OOV recall rate tells us how many OOV words are covered by the word-formation rules. The wordlist used in running the scoring tool consists of all the 89,845 entries in our dictionary.

In the evaluation, we first segmented the text using our default setting where every parameter was set to 0. This gave us the maximal word in each case. We then did a quick resetting of the parameters following the relevant guidelines. Results of both the default segmentation and the adjusted segmentation were evaluated against the CHTB and BU gold standards respectively. The differences between the default setting scores and the scores after parameter value adjustment thus reflect the amount of customization that has been achieved:

When evaluated against the CHTB gold standard, our system received the following scores when the default setting was used:

Word Recall:	83.4 %
Word Precision:	90.1%
F-measure:	86.6%
OOV Rate:	8.4%
OOV Recall Rate:	58.8%

After a quick adjustment, during which 19 parameters were reset from 0 to 1, the scores became:

Word Recall:	96.5%
Word Precision:	96.3%
F-measure:	96.4%
OOV Rate:	8.4%
OOV Recall Rate:	86.5%

When evaluated against the BU gold standard, our system received the following scores when the default setting was used:

Word Recall:	84.4%
Word Precision:	90.4%
F-measure:	87.3%
OOV Rate:	7.5%
OOV Recall Rate:	49.2%

After the resetting of 22 parameters from 0 to 1, the scores became

Word Recall:	96.8%
Word Precision:	95.9%
F-measure:	96.3%
OOV Rate:	7.5%
OOV Recall Rate:	81.1%

We see that the scores improved dramatically across the board in both the CHTB and BU data after the parameter values were adjusted to the relevant standards. In particular, there is a high correlation between the rise of OOV Recall Rate and the F-measure, which indicates that the improvements indeed came from the area of MDWs.

We also tried the setting where every parameter was set to 1, which resulted in the display of minimal words. Here are the scores:

CHTB:	Word Recall:	86.4 %
	Word Precision:	78.6 %
	F-measure:	82.3 %
	OOV Rate:	8.4 %
	OOV Recall Rate:	12.7 %

BU:	Word Recall:	91.8 %
	Word Precision:	86.1%
	F-measure:	88.9 %
	OOV Rate:	7.5 %
	OOV Recall Rate:	21.9 %

This is the result we would get if we depended only on our dictionary and no MDWs rules were applied. The scores dropped sharply in both the CHTB and BU cases. Of particular interest is the drop in the OOV recall rates. If all the OOV words were constructed by MDW rules, the OOV recall rate would be 0 when we display the minimal words, which are all in the dictionary. However, there are other processes in our system that assemble dictionary words into bigger units and these units are invariably displayed as single words. For example, "1978" always appears as a single word in spite of the fact that it is assembled from "1", "9", "7" and "8" at run time. Another example is English words in Chinese texts, such as "IBM" which is not in our dictionary. MDWs thus account for 85.8% of the OOV recall rate in CHTB and 73.1% of the OOV recall rate in BU.

The evaluation results show clearly that (1) the variation among different standards does come largely from the area of MDWs and (2) our system can adapt to different standards successfully by parameterizing the display of MDWs.

3.4 Customizable resources

So far we have focused on the customization of a single segmentation system to produce different outputs. We can also envision an approach where segmenters for different standards are built by training them on texts that have been segmented according to those standards. This leads to the question of whether we can develop language resources that can be customized to serve different purposes. The annotated corpora that are currently being developed in the Chinese NLP community mostly follow a single standard and they are usually not designed for the training of segmenters that do not follow the same standard. However, we cannot afford to build a different tagged corpus for each different standard. It will be highly desirable, therefore, to develop resources that are customizable. The requirement for segmented texts, then, is that it should be capable of being converted to segmentations of varying granularity. To achieve this goal, we have to tag our texts in such a way that (1) the internal structures of words (at least the MDWs) are represented and (2) word boundaries of different types can be selectively kept or removed with ease.

Certain word-internal structures are already preserved in some annotated corpora. In

CHTB, for example, verbs and their directional/resultative complements are grouped into single units with internal word boundaries. 走进去 is thus tagged as "(走 进去)" and 走不进去 as "(走 不 进去)". The bracketing of named entities in the BU corpora is another step in this direction. The ROCLING standard has set even higher goals. It classifies segmentation into three increasingly demanding levels: faithful (信 xin), truthful (达 da) and graceful (雅 ya) [Huang *et al*. 1997].[16] The segmentation units at the faithful level basically correspond to the minimal words in our system. Those at the truthful level are usually MDWs. Segmentation units at the graceful level are not as well defined, but some of them correspond to the maximal words in our system, such as company names. Units at these levels are to be tagged with different SGML tags: faithful-level words tagged as <w0>, truthful-level words tagged as <w1>, and graceful-level words tagged as <w2>. "赵元任语言学基金会" will probably be tagged as the following in this scheme, assuming 赵, 元 and 任 are in the dictionary but 赵元任 and 元任 are not:

<w2>

<w1> <w0>赵</w0> <w0>元</w0> <w0>任</w0> </w1>

<w1> <w0>语言</w0> <w0>学</w0> </w1>

<w1> <w0>基金</w0> <w0>会</w0> </w1>

</w2>

This tagging scheme makes the tagged data customizable, since all the potential word boundaries are preserved. But it does not distinguish between different types of MDWs and therefore the choices for customization are more limited. To preserve the type information of MDWs, we will need the following representation:

```
<OrgName>
   <FullName>
      <Char>赵</ Char >
      <GivenName> < Char >元</ Char > < Char >任</ Char > < /GivenName >
   </FullName >
   <OrgName>
      <NounSfx> <Noun>语言</ Noun > <Suffix>学</ Suffix > </ NounSfx >
      < NounSfx > < Noun >基金</ Noun > < Suffix >会</ Suffix > </ NounSfx >
   </OrgName>
</ OrgName >
```

This representation is equivalent to the word tree in Figure 1. It is somewhat clumsy,

[16] (a) Faithful (信 xin): All segmentation units listed in the reference lexicon should be successfully segmented; (b)Truthful (达 da): In addition to (a), all segmentation units derivable by morphological rules should be successfully segmented; Graceful (雅 ya): Segmentation units are ideal linguistic words for fully automated language understanding.

however, and may not be optimal when it comes to large-scale tagging. A simpler representation might be:

赵<3>元<4>任<1>语言<2>学<1>基金<2>会

where each number corresponds to a label, namely 1 = OrgName, 2 = NounSfx, 3 = Fullname, and 4 = GivenName. Since each label represents the morphological rule that assembles the pieces into a single unit, we replace each word-internal boundary with the relevant number that corresponds to the rule that puts the pieces together. We can then obtain different segmentations by specifying the types of boundaries to be kept or removed. During customization, the boundaries to be kept will be replaced by spaces and the ones to be removed will disappear. In the above example, if we want to treat personal names and words derived from suffixation as single words while keeping components of an organization name apart, we can remove <2>, <3> and <4> and turn the other numbers into spaces. The result will be "赵元任 语言学 基金会". We will get "赵 元任 语言 学 基金 会" if the number to be removed is just 4. It should be noted that, just like the case of parameter setting in our system, not all the number combinations are possible in the replacement/removal. For example, we cannot remove <1> and replace all the other numbers with spaces, since we cannot keep the whole organization name as a single piece if we break up its components. Therefore, there need to be a partial order of those numbers where the removal of a given number implies the removal of some other numbers. The original motivation of this representation was to avoid the need to process the same text *N* times to get *N* different segmentations. We were able to process the corpus just once and use the same output for multiple purposes. It seems that this can be an option in the future development of Chinese language resources.

In principle, all the information represented in the word trees of our system can be represented in a tagged corpus. In practice, however, textual representation of certain information (e.g. the lemma attribute) can be cumbersome and it can be labor-intensive for the annotators. Besides, the tagging is not easy to change once it is done. The main advantage of a customizable system over a customizable corpus is that the former can adapt to new specifications of representation very quickly, with large-scale systematic changes made within a very short time. This is especially so in cases of "bracketing paradoxes" where incompatible representations might have to be generated for different purposes. Of course, the output of an automatic system may be inferior in accuracy to a hand-tagged corpus, but we can maintain a set of surface sentences which are known to have the correct output from the system. Every time the "spec" changes, we can modify the system and process those sentences again to produce the updated output instead of modifying the whole tagged corpus.

3.5 Future refinement

In our current implementation of the multi-resolution parameters, the parameter values are not probabilistic in nature. They are either 0 or 1 and therefore it is not able to make the finer distinctions that we sometimes need when we try to determine wordhood on the basis of statistical information. As we have seen in Section 2, the segmentation of certain MDWs can depend on the frequency of those MDWs and the mutual information between their components. To make our customization more fine-tuned, we need to take such probabilistic information into account. One way to do it is to gather statistical information for every MDW and normalize it into a value between 0 and 1. This value can then be combined with the parameter values that we set by hand to produce a probability that represents the likelihood of a MDW being broken into individual words. We can then set a threshold to determine the "resolution" of the segmentation.

4. Conclusion

The standards for Chinese word segmentation can vary according to different definitions of words and the different requirements of NLP applications. It is therefore important that the segmentation systems we develop or the tagged corpora we construct be capable of being customized to meet different needs. In this paper, we have concentrated on the segmentation of morphologically derived words (MDWs). We have demonstrated that a segmentation system can be customized to produce different outputs for different standards if the word-internal structures of MDWs are preserved in a tree structure and different types of nodes in the tree are associated with different resolution parameters. Different settings of those parameters then result in segmentations of different granularities. Evaluation shows that the effect of customization is significant and MDWs are indeed the main area where customization is most needed. A similar approach can also be used in the development of linguistic resources where a single annotated corpus can be customized to provide training and testing data for different applications.

References

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

Dai, J. X.-L, "Syntactic, morphological and phonological words in Chinese", in Packard (1997), pp. 103-134.

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

Di Sciullo, A. M. and E. Williams, *On the Definition of Word*. MIT Press, Cambridge, MA, 1987.

Duanmu, S., "Wordhood in Chinese". In Packard (1997) pp.135-196.

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

Heidorn, G. E., "Intelligent writing assistance", in *A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text*, Dale R., Moisl H., and Somers H. eds., Marcel Dekker, New York, 2000, pp. 181-207.

Huang, C., K. Chen, F. Chen and L. Chang, Segmentation standard for Chinese natural language processing. *International Journal of Computational Linguistics and Chinese Language Processing*, 2(2), 1997, pp. 47-62.

Jensen, K., G. Heidorn and S. Richardson. *Natural Language Processing: the PLNLP Approach"*. Kluwer Academic Publishers, Boston, 1993.

Packard, J. (ed.), *New Approaches to Chinese Word Formation: Morphology, phonology and the lexicon in modern and ancient Chinese. Trends in Linguistics Studies and Monographs 105*. Mouton de Gruyter, Berlin and New York, 1997.

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

ROCLING Segmentation Principle for Chinese Language Processing, 1997, http://godel.iis.sinica.edu.tw/ROCLING/juhuashu1.htm

Sadock, J.M., *Autolexical Syntax*. University of Chicago Press, Chicago, 1991.

Selkirk, E., *The Syntax of Words*. The MIT Press, Cambridge, MA, 1982.

Sproat, R., Corpus-Based Methods in Chinese Morphology. Tutorial at the 19th International Conference on Computational Linguistics, 2002.

Sproat, R., *A Computational Theory of Writing Systems*. Cambridge University Press, Stanford, CA, 2000.

Sproat, R., C. Shih, W. Gale and N. Chang, "A stochastic finite-state word-segmentation algorithm for Chinese". *Computational Linguistics*, 22(3), 1996, pp. 377-404.

Sproat, R. and C. Shih, "A statistical method for finding word boundaries in Chinese text", *Computer Processing of Chinese and Oriental Languages*, Vol. 4, 1990, pp. 336-351.

Wu, A. and Z. Jiang, "Statistically-Enhanced New Word Identification in a Rule-based Chinese System". In *Proceedings of the Second ACL Chinese Processing Workshop*, HKUST, Hong Kong, 2000, pp. 46-51.

Wu A. and Z. Jiang, "Word Segmentation in Sentence Analysis". In *Proceedings of the 1998 International Conference on Chinese Information Processing*, Beijing, China, 1998, pp. 169-180.

Xia, F., The Segmentation Guidelines for the Penn Chinese Treebank (3.0). Technical report, University of Pennsylvania, 2000, http://www.cis.upenn.edu/~chinese/.

Yu, S., Guidelines for the Annotation of Contemporary Chinese Texts: word segmentation and POS-tagging, Institute of Computational Linguistics, Beijing University, Beijing, 1999

Appendix

I. Examples of reduplication

- AA
  看看 kan-kan: look-look "*take a look*"
  红红 hong-hong: red-red "*very red / kind of red*"
  慢慢 man-man: slow-slow "*slowly*"
  年年 nian-nian: year-year "*every year*"
- ABAB
  研究研究 yanjiu-yanjiu: research-research "*do some research*"
  舒服舒服 shufu-shufu: comfortable-comfortable "*have a comfortable time*"
- AABB
  方方面面 fang-fang-mian-mian "*every aspect*"
  清清楚楚 qing-qing-chu-chu "*very clear*"
  痛痛快快 tong-tong-kuai-kuai "*thoroughly*"
  年年月月 nian-nian-yue-yue: year-year-month-month "*year after year, month after month*"
- AXA
  试一试 shi-yi-shi: try-one-try "*give it a try*"
  试了试 shi-le-shi: try-LE-try "*gave it a try*"
  试了一试 shi-le-yi-shi: try-LE-one-try "*gave it a try*"
- AXAY
  跑来跑去 pao-lai-pao-qu: run-come-run-go "*run around*"
  送医送药 song-yi-song-yao:send-doctor-send-medicine "*deliver medical aid*"
  一砖一瓦 yi-zhuan-yi-wa: one-brick-one-tile "*every brick / brick by brick*"
  所言所行 suo-yan-suo-xing: SUO-speak-SUO-do "*every word and deed*"
- XAYA
  东看西看 dong-kan-xi-kan: east-look-west-look "*look here and there*"
  左挑右挑 zuo-tiao-you-tiao: pick-left-pick-right "*pick and choose*"
- AA 看
  试试看 shi-shi-kan: try-try-look "*give it a try*"
- AAB
  充充电 chong-chong-dian: "*charge the battery a bit*"
  溜溜光 liu-liu-guang "*very smooth*"
- ABB
  亮堂堂 liang-tang-tang "*very bright* "

II. Examples of Derivational Affixation

1. Prefixation

- Prefix + Noun => Noun
  微电子 wei-dianzi "*micro-electronics*"
- Prefix + Noun => Adj
  防病毒（软件） fang-bingdu "*anti-virus*"
- Prefix + Verb => Adj
  可再生（能源） ke-zaisheng "*re-usable*"

- Prefix + Number => Number
  第一 di-yi "*first*"

2. Suffixation

- Noun + Suffix => Noun
  科学家 kexue-jia "*scientist*"
- Noun + Suffix => Adj
  意大利式 yidali-shi "*Italian-style*"
- Verb + Suffix => Noun
  邮递员 youdi-yuan "*mail-man*"
- Verb + Suffix => Adj
  渐进式 jianjin-shi "*gradual-mode*"
- Adj + Suffix => Noun
  积极性 jiji-xing "*proactive-ness*"

Computational Linguistics and Chinese Language Processing
Vol. 8, No. 1, February 2003, pp.29-48

Chinese Word Segmentation as Character Tagging

Nianwen Xue*

Abstract

In this paper we report results of a supervised machine-learning approach to Chinese word segmentation. A maximum entropy tagger is trained on manually annotated data to automatically assign to Chinese characters, or *hanzi*, tags that indicate the position of a *hanzi* within a word. The tagged output is then converted into segmented text for evaluation. Preliminary results show that this approach is competitive against other supervised machine-learning segmenters reported in previous studies, achieving precision and recall rates of 95.01% and 94.94% respectively, trained on a 237K-word training set.

Key Words: Chinese word segmentation, supervised machine-learning, maximum entropy, character tagging

1. Introduction

It is generally agreed among researchers that word segmentation is a necessary first step in Chinese language processing. However, unlike English text in which sentences are sequences of words delimited by white spaces, in Chinese text, sentences are represented as strings of Chinese characters or *hanzi* without similar natural delimiters. Therefore, the first step in a Chinese language processing task is to identify the sequence of words in a sentence and mark boundaries in appropriate places. This may sound simple enough but in reality identifying words in Chinese is a non-trivial problem that has drawn a large body of research in the Chinese language processing community [Fan and Tsai, 1988; Gan, 1995; Gan, Palmer, and Lua, 1996; Guo, 1997; Jin and Chen, 1998; Sproat and Shih, 1990; Sproat *et al*., 1996; Wu and Jiang, 1998; Wu, 2003].

It is easy to demonstrate that the lack of natural delimiters itself is not the heart of the problem. In a hypothetical language where all words are represented with a finite set of symbols, if one subset of the symbols always start a word and another subset, mutually exclusive from the previous subset, always end a word, identifying words would be a trivial

* Institute for Research in Cognitive Science , Suite 400A, 3401 Walnut Street
University of Pennsylvania, Philadelphia, PA 19104, USA
E-mail: xueniwen@linc.cis.upenn.edu

exercise. Nor can the problem be attributed to the lack of inflectional morphology. Although it is true in Indo-European languages inflectional affixes can generally be used to signal word boundaries, it is conceivable that a hypothetical language can use symbols other than inflectional morphemes to serve the same purpose. Therefore the issue is neither the lack of natural word delimiters nor the lack of inflectional morphemes in a language, rather it is whether the language has a way of unambiguously signaling the boundaries of a word.

The real difficulty in automatic Chinese word segmentation is the lack of such unambiguous word boundary indicators. In fact, most *hanzi* can occur in different positions within different words. The examples in Table 1 show how the Chinese character 产 ("produce") can occur in four different positions. This state of affairs makes it impossible to simply list mutually exclusive subsets of *hanzi* that have distinct distributions, even though the number of *hanzi* in the Chinese writing system is in fact finite. As long as a *hanzi* can occur in different word-internal positions, it cannot be relied upon to determine word boundaries as they could be if their positions were more or less fixed.

Table 1. A* hanzi *can occur in multiple word-internal positions

Position	Example
Left	产生 'to come up with'
Word by itself	产小麦 'to grow wheat'
Middle	生产线 'assembly line'
Right	生产 'to produce'

The fact that a *hanzi* can occur in multiple word-internal positions leads to ambiguities of various kinds, which are described in detail in [Gan, 1995]. For example, 文 can occur in both word-initial and word-final positions. It occurs in the word-final position in 日文 ("Japanese") but in the word-initial position in 文章("article"). In a sentence that has a string "日文章", as in (1)[1], an automatic segmenter would face the dilemma whether to insert a word boundary marker between 日 and 文, thus grouping 文章 as a word, or to mark 日文 as a word, to the exclusion of 章. The same scenario also applies to 章, since like 文, it can also occur in both word-initial and word-final positions.

1. (a) Segmentation I

日文 章魚 怎麼 說?

Japanese octopus how say

"How to say octopus in Japanese?"

(b) Segmentation II

[1]Adapted from [Sproat *et al*.,1996]

日 文章 魚 怎麼 說?

Japan article fish how say

Ambiguity also arises because some *hanzi* should be considered to be just word components in certain contexts and words by themselves in others. For example, 魚 can be considered to be just a word component in 章魚. It can also be a word by itself in other contexts. Presented with the string 章魚 in a Chinese sentence, a human or automatic segmenter would have to decide whether 魚 should be a word by itself or form another word with the previous *hanzi*. Given that 日, 文章, 章魚, 魚 are all possible words in Chinese, how does one decide that 日文 章魚 is the right segmentation for the sentence in (1) while 日 文章 魚 is not? Obviously it is not enough to know just what words are in the lexicon. In this specific case, a human segmenter can resort to world knowledge to resolve this ambiguity, knowing that 日 文章 魚 would not make any kind of real-world sense.

In other cases a human segmenter can also rely on syntactic knowledge to properly segment a sentence. For instance, 枪 should be considered a word in (2a) and two words in (2b):

2. a 警察 枪-杀 了 那 个 逃犯

police gun-kill LE that CL escapee

"Police killed the escapee with a gun."

b 警察 用 枪 杀 了 那 个 逃犯

Police with gun kill LE that CL escapee

"Police killed the escapee with a gun"

In (2b), 枪 is a word by itself and forms a phrasal constituent with the preceding 用. In order to get the segmentation right for the example in (2) one needs to know, for example, that 用 has to take a complement and in the case of (2b) the complement is 枪. Therefore it is impossible for 枪 to be part of the word 枪杀. The human segmenter has little difficulty resolving these ambiguities and coming up with the correct segmentation since they have linguistic and world knowledge at their disposal. However, the means available to the human segmenter cannot be made available to computers just as easily. As a result, an automatic word segmenter would have to bypass such limitations to resolve these ambiguities.

In addition to the ambiguity problem, another problem that is often cited in the literature is the problem of so-called out-of-vocabulary or "unknown" words [Wu and Jiang, 1998]. The unknown word problem arises because machine-readable dictionaries cannot possibly list all

the words encountered in NLP tasks exhaustively[2]. For one thing, although the number of *hanzi* generally remains constant, Chinese has several productive new word creation mechanisms. First of all, new words can be created through compounding, in which new words are formed through the combination of existing words, or through *suoxie*, in which components of existing words are extracted and combined to form new words. Second, new names are created by combining existing characters in a very unpredictable manner. Third, there are also transliterations of foreign names. These are just a few of the many ways new words can be introduced in Chinese.

The key to accurate automatic word identification in Chinese lies in the successful resolution of these ambiguities and a proper way to handle out-of-vocabulary words. We have demonstrated that the ambiguities in Chinese word segmentation is due to the fact that a *hanzi* can occur in different word-internal positions. Given the proper context, generally provided by the sentence in which it occurs, the position of a *hanzi* can be determined. If the positions of all the *hanzi* in a sentence can be determined with the help of the context, the word segmentation problem would be solved. This is the line of thinking we are going to pursue in the present work. There are several reasons why we may expect this approach to work. First, Chinese words generally have fewer than four characters. As a result, the number of positions is small. Second, although each *hanzi* can in principle occur in all possible positions, not all *hanzi* behave this way. A substantial number of *hanzi* are distributed in a constrained manner. For example, 们, the plural marker, almost always occurs in the word-final position. Finally, although Chinese words cannot be exhaustively listed and new words are bound to occur in naturally occurring text, the same is not true for *hanzi*. The number of *hanzi* stays fairly constant and we do not generally expect to see new *hanzi*. In this paper, we model the Chinese word segmentation problem as a *hanzi* tagging problem and use a machine-learning algorithm to determine the word-internal positions of *hanzi* with the help of contextual information.

The remainder of this paper is organized as follows. In Section 2, we briefly review the representative approaches in the previous studies on Chinese word segmentation. In Section 3, we describe how the word segmentation problem can be modeled as a tagging problem and how the maximum entropy model is used to solve this problem. We describe our experiments in Section 4. In Section 5, we report our experimental results, using the maximum matching algorithm as a baseline. We also evaluate these results against previous approaches and discuss the contributions of different feature sets and the effectiveness of different tag sets. We conclude this paper and discuss future work in Section 6.

[2]See [Guo, 1997] for a different point of view

2. Previous Work

Various methods have been proposed to address the word segmentation problem in previous studies. Noting that linguistic information, syntactic information in particular, can help identify words, [Gan, 1995] and [Wu and Jiang, 1998] treated word segmentation as inseparable from Chinese sentence understanding as a whole. As a result, the success of the word segmentation task is tied to the success of the sentence understanding task, which is just as difficult as the word segmentation problem, if not more difficult. Most of the word segmentation systems reported in previous studies are stand-alone systems and they fall into three main categories, depending on whether they use statistical information and electronic dictionaries. These are purely statistical approaches [Sproat and Shih, 1990; Sun, Shen, and Tsou, 1998; Ge, Pratt, and Smyth, 1999; Peng and Schuurmans, 2001], non-statistical dictionary-based approaches [Liang, 1993; Gu and Mao, 1994] and statistical and dictionary-based approaches [Sproat *et al*., 1996]. More recently work on Chinese word segmentation also includes supervised machine-learning approaches [Palmer, 1997; Hockenmaier and Brew, 1998; Xue, 2001].

Purely dictionary-based approaches generally addresses the ambiguity problem with some heuristics, and the most successful heuristics are variations of the maximum matching algorithm. A maximum matching algorithm is a greedy search routine that walks through a sentence trying to find the longest string of *hanzi* starting from a given point in the sentence that matches a word entry in a pre-compiled dictionary. For instance, assuming 关 ("close"), 心 ("heart") and 关心 ("care about") are all listed in the dictionary, given a string of *hanzi* 关-心, the maximum matching algorithm always favors 关心 as a word, over 关-心 as a string of two words. This is because 关心 is a longer string than 关 and both of them are in the dictionary. When the segmenter finds 关, it will continue to search and see if there is a possible extension. When it finds another word 关心 in the dictionary it will decide against inserting a word boundary between 关 and 心. When the algorithm can no longer extend the string of *hanzi* it stops searching and inserts a word boundary marker. The process is repeated from the next *hanzi* till it reaches the end of the sentence. The algorithm is successful because in a lot of cases, the longest string also happens to be correct segmentation. For example, for the example in (1), the algorithm will rightly decide that (1a) rather than (1b) is the correct segmentation for the sentence, assuming 日, 日文, 文章, 章鱼 and 鱼 are all listed in the dictionary. However, this algorithm will output the wrong segmentation for (2b), in which it will incorrectly group 枪杀 as a word. In addition, the maximum matching algorithm does not have a built-in mechanism to deal with out-of-vocabulary words. In general, the completeness of the dictionary to a large extent determines the degree of success for segmenters using this approach.

As a representative of purely statistical approaches, [Sproat and Shih, 1990] relies on the mutual information of two adjacent characters to decide whether they form a two-character word. Given a string of characters $c_1...c_n$, the pair of adjacent characters with the largest mutual information greater than a pre-determined threshold is grouped as a word. This process is repeated until there are no more pairs of adjacent characters with a mutual information value greater than the threshold. This algorithm is extended by [Sun, Shen, and Tsou, 1998] so that association measures other than mutual information are also taken into consideration. More recently, [Ge, Pratt, and Smyth, 1999; Peng and Schuurmans, 2001] applied expectation maximization methods to Chinese word segmentation. For example, [Peng and Schuurmans, 2001] used an EM-based algorithm to estimate probabilities for words in a dictionary and use mutual information to weed out proposed words whose components are not strongly associated. Purely statistical approaches have the advantage of not needing a dictionary or training data, and since unsegmented data are easy to obtain, they can be easily trained on any data source. The drawback is that statistical approaches generally do not perform well in terms of the accuracy of the segmentation.

Statistical dictionary-based approaches attempt to get the best of both worlds by combining the use of a dictionary and statistical information such as word frequency. [Sproat *et al*., 1996] represents a dictionary as a weighted finite-state transducer. Each dictionary entry is represented as a sequence of arcs labeled with a *hanzi* and its phonemic transcription, starting from an initial state 0 and terminated by a *weighted* arch labeled with an empty string ε and a part-of-speech tag. The weight represents the estimated cost of the word, which is its negative log probability. The probabilities of the dictionary words as well as morphologically derived words not in the dictionary are estimated from a large unlabeled corpus. Given a string of acceptable symbols (all the *hanzi* plus the empty string), there exists a function that takes this string of symbols as input and produces as output a transducer that maps all the symbols to themselves. The path that has the cheapest cost is selected as the best segmentation for this string of characters. Compared with purely statistical approaches, statistical dictionary-based approaches have the guidance of a dictionary and as a result they generally outperform purely statistical approaches in terms of segmentation accuracy.

Recent work on Chinese word segmentation has also used the transformation-based error-driven algorithm [Brill, 1993] and achieved various degrees of success [Palmer, 1997; Hockenmaier and Brew, 1998; Xue, 2001]. The transformation-based error-driven algorithm is a supervised machine-learning routine first proposed by [Brill, 1993] and initially used in POS tagging as well as parsing. It has been applied to Chinese word segmentation by [Palmer, 1997; Hockenmaier and Brew, 1998; Xue, 2001]. Although the actual implementation of this algorithm may differ slightly, in general the transformation-based error-driven approaches try

to learn a set of n-gram rules from a training corpus and apply them to segment new text. The input to the learning routine is a (manually or automatically) segmented corpus and its unsegmenteded (or undersegmented) counterpart. The learning algorithm compares the segmented corpus and the undersegmented dummy corpus at each iteration and finds the rule that achieves the maximum gain if applied. The rule with the maximum gain is the one that makes the dummy corpus most like the reference corpus. The maximum gain is calculated with an evaluation function which quantifies the gain and takes the largest value. The rules are instantiations of a set of pre-defined templates. After the rule with the maximum gain is found, it is applied to the dummy corpus, which will better resemble the reference corpus as a result. This process is repeated until the maximum gain drops below a pre-defined threshold, which indicates improvement achieved through further training will no longer be significant. The output of the training process would be a ranked set of rules instantiating the predefined set of templates. The rules will then be used to segment new text. Like statistical approaches, this approach provides a trainable method to learn the rules from a corpus and it is not labor-intensive. The drawback is that compared with statistical approaches, this algorithm is not very efficient.

The present work represents another supervised machine-learning approach. Specifically, we applied the maximum entropy model, a statistical machine-learning algorithm to Chinese word segmentation.

3. A supervised machine-learning algorithm to Chinese word segmentation

In this section, we first formalize the idea of tagging *hanzi* based on their word-internal positions and describe the tag set we used. We then briefly describe the maximum entropy model, which has been successfully applied to POS tagging as well as parsing [Ratnaparkhi, 1996; Ratnaparkhi, 1998].

3.1 Reformulating word segmentation as a tagging problem

Before we apply the machine-learning algorithm first we convert the manually segmented words in the corpus into a tagged sequence of Chinese characters. To do this, we tag each character with one of the four tags, LL, RR, MM and LR depending on its position within a word. It is tagged LL if it occurs on the left boundary of a word, and forms a word with the character(s) on its right. It is tagged RR if it occurs on the right boundary of a word, and forms a word with the character(s) on its left. It is tagged MM if it occurs in the middle of a word. It is tagged LR if it forms a word by itself. We call such tags position-of-character (POC) tags to differentiate them from the more familiar part-of-speech (POS) tags. For example, the manually segmented string in (3a) will be tagged as (3b):

3. (a) 上海 计划 到 本 世纪 末 实现 人均 国内 生产 总值 五千 美元

(b) 上/LL 海/RR 计/LL 划/RR 到/LR 本/LR 世/LL 纪/RR 末/LR 实/LL 现/RR 人/LL 均/RR 国/LL 内/RR 生/LL 产/RR 总/LL 值/RR 五/LL 千/RR 美/LL 元/RR

(c) Shanghai plans to reach the goal of 5,000 dollars in per capita GDP by the end of the century.

Given a manually segmented corpus, a POC-tagged corpus can be derived trivially with perfect accuracy. The reason why we use such POC-tagged sequences of characters instead of applying n-gram rules to segmented corpus directly [Palmer, 1997; Hockenmaier and Brew, 1998; Xue, 2001] is that they are much easier to manipulate in the training process. In addition, the POC tags reflect our observation that the ambiguity problem is due to the fact that a *hanzi* can occur in different word-internal positions and it can be resolved in context. Naturally, while some characters have only one POC tag, most characters will receive multiple POC tags, in the same way that words can have multiple POS tags. Table 2 shows how all four of the POC tags can be assigned to the character 产 ("produce"):

Table2. A character can receive as many as four tags

Position	Tag	Example
Left	LL	产生 'to come up with'
Word by itself	LR	产 小麦 'to grow wheat'
Middle	MM	生产线 'assembly line'
Right	RR	生产 'to produce'

If there is ambiguity in segmenting a sentence or any string of *hanzi*, then there must be some *hanzi* in the sentence that can receive multiple tags. For example, each of the first four characters of the sentence in (1) would have two tags. The task of the word segmentation is to choose the correct tag for each of the *hanzi* in the sentence. The eight possible tag sequences for (1) are shown in (4a), and the correct tag sequence is (4b).

4. (a) 日/LL|LR 文/RR|LL 章/LL|RR 鱼/RR|LR 怎/LL 么/RR 说/LR ?

(b) 日/LL 文/RR 章/LL 鱼/RR 怎/LL 么/RR 说/LR ?

Also like POS tags, how a character is POC-tagged in naturally occurring text is affected by the context in which it occurs. For example, if the preceding character is tagged LR or RR, then the next character can only be tagged LL or LR. How a character is tagged is also affected by the surrounding characters. For example, 关 (``close'') should be tagged RR if the previous character is 开 (``open'') and neither of them forms a word with other characters, while it should be tagged LL if the next character is 心 (``heart'') and neither of them forms a word with other characters. This state of affairs closely mimics the familiar POS tagging

problem and lends itself naturally to a solution similar to that of POS tagging. The task is one of ambiguity resolution in which the correct POC tag is determined among several possible POC tags in a specific context. Our next step is to train a maximum entropy model on the perfectly POC-tagged data derived from a manually segmented corpus to automatically POC-tag unseen text.

3.2 The maximum entropy tagger

The maximum entropy model used in POS-tagging is described in detail in [Ratnaparkhi, 1996] and the POC tagger here uses the same probability model. The probability model is defined over $H \times T$, where H is the set of possible contexts or "histories" and T is the set of possible tags. The model's joint probability of a history h and a tag t is defined as

$$p(h,t) = \pi\mu\prod_{j=1}^{k}\alpha_j^{f_j(h,t)} \tag{1}$$

where π is a normalization constant, $\{\mu,\alpha_1,...,\alpha_k\}$ are the model parameters and $\{f_1,...,f_k\}$ are known as features, where $f_j(h,t)\in\{0,1\}$. Each feature f_j has a corresponding parameter α_j, hat effectively serves as a "weight" of this feature. In the training process, given a sequence of characters $\{c_1,...,c_n\}$ and their POC tags $\{t_1,...,t_n\}$ as training data, the purpose is to determine the parameters $\{\mu,\alpha_1,...,\alpha_k\}$ that maximize the likelihood of the training data using p:

$$L(P) = \prod_{i=1}^{n}P(h_i,t_i) = \prod_{i=1}^{n}\pi\mu\prod_{j=1}^{k}\alpha_j^{f_j(h_i,t_i)} \tag{2}$$

The success of the model in tagging depends to a large extent on the selection of suitable features. Given (h,t), a feature must encode information that helps to predict t. The features we used in this experiment are instantiations of the feature templates in (5). Feature templates (b) to (e) represent character features while (f) represents tag features. The character and tag features are also represented graphically in Figure 1, where $C_{-3}...C_3$ are characters and $T_{-3}...T_3$ are POC tags. Each arrow or arc represents one feature template. Feature template (a) represents the default feature.

5 Feature templates

(a) Default feature

(b) The current character (C_0)

(c) The previous (next) two characters (C_{-2}, C_{-1}, C_1, C_2)

(d) The previous (next) character and the current character ($C_{-1}\,C_0$, $C_0\,C_1$),

the previous two characters (C_{-2} C_{-1}), and

the next two characters (C_1 C_2)

(e) The previous and the next character (C_{-1} C_1)

(f) The tag of the previous character (T_{-1}), and
the tag of the character two before the current character (T_{-2})

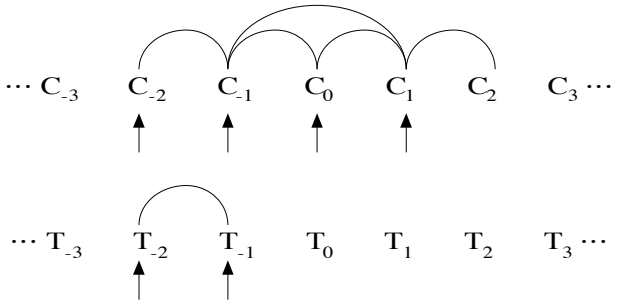

Figure 1 Features used in the maximum entropy segmenter

In general, given (h,t), these features are in the form of co-occurrence relations between t and some type of context h, or between t and some properties of the current character. For example,

$$f_i(h_i,t_i) = \begin{cases} 1 & \text{if } t_{i-1}=LL \ \& \ t_i=RR \\ 0 & \text{otherwise} \end{cases}$$

This feature will map to 1 and contribute towards $p(h_i,t_i)$ if $c_{(i-1)}$ is tagged LL and c_i is tagged RR.

The feature templates in (5) encode three types of contexts. First, features based on the current and surrounding characters (5b, 5c, 5d, 5e) are extracted. Given a character in a sentence, this model will look at the current character, the previous two and next two characters. For example, if the current character is 们 (plural marker), it is very likely that it will occur as a suffix in a word, thus receiving the tag RR. On the other hand, for other characters, they might be equally likely to appear on the left, on the right or in the middle. In those cases where it occurs within a word depends on its surrounding characters. For example, if the current character is 爱 ("love"), it should perhaps be tagged LL if the next character is 护 ("protect"). However, if the previous character is 热 ("warm"), then it should perhaps be tagged RR. Second, features based on the previous tags (5f) are extracted. Information like this is useful in predicting the POC tag for the current character just as the POS tags are useful in predicting the POS tag of the current word in a similar context. For example, if the previous character is tagged LR or RR, this means that the current character must start a word, and

should be tagged either LL or LR. Finally, a default feature (5a) is used to capture cases where no other features are available. When the training is complete, the features and their corresponding parameters will be used to calculate the probability of the tag sequence of a sentence when the tagger tags unseen data. Given a sequence of characters $\{c_1,...,c_n\}$, the tagger searches for the tag sequence $\{t_1,...,t_n\}$ with the highest probability

$$P(t_1,...t_n \mid C_1,...C_n) = \prod_{i=1}^{n} P(t_i \mid h_i) \tag{3}$$

and the conditional probability of for each POC tag t given its history h is calculated as

$$P(t \mid h) = \frac{p(h,t)}{\sum_{t' \in T} P(h,t')} \tag{4}$$

4. Experiments

We conducted two experiments. In the first experiment, we used the maximum matching algorithm to establish a baseline, as comparing results across different data sources can be difficult. This experiment is also designed to test the performance of the maximum matching algorithm with or without unknown words. In the second experiment, we applied the maximum entropy model to the problem of Chinese word segmentation. The data we used is from the Penn Chinese Treebank [Xia *et al*., 2000; Xue, Chiou, and Palmer, 2002] and it consists of Xinhua newswire articles. We took 250,389-word (426,292 characters or hanzi) worth of manually segmented data and divided them into two chunks. The first chunk has 237,791 words (404,680 Chinese characters) and is used as training data. The second chunk has 12,598 words (21,612 characters) and is held out as testing data. This data is used in both of our experiments.

4.1 Experiment One

In this experiment, we conducted two sub-experiments. In the first sub-experiment, we used a forward maximum matching algorithm to segment the testing data with a dictionary compiled from the training data. There are 497 (or 3.95%) new words (words that are not found in the training data) in the testing data. In the second sub-experiment, the same algorithm was used to segment the same testing data with a dictionary compiled from BOTH the training data and the testing data. In other words, there is no new word in the testing data.

4.2 Experiment Two

In the second experiment, a maximum entropy model was trained on a POC-tagged corpus

derived from the training data described above. In the testing phase, the sentences in the testing data were first split into sequences of *hanzi* and then tagged with this maximum entropy tagger. The tagged testing data is then converted back into word segments for evaluation. Note that converting a POC-tagged corpus into a segmented corpus is not entirely straightforward when inconsistent tagging occurs. For example, it is possible that the tagger assigns a LL-LR sequence to two adjacent characters. We made no effort to ensure the best possible conversion. The character that is POC-tagged LL is invariably combined with the following character, no matter how the latter is tagged. The example in (6) illustrates this process.

6. (a) Tagged output

在/LR 刚/LL 刚/RR 过/LL 去/RR 的/LR 一/LL 九/MM 九/MM 七/MM 年/RR ，/LR 中/LL 国/RR 进/LL 出/MM 口/RR 贸/LL 易/RR 中/LR ，/LR 国/LL 有/RR 企/LL 业/RR 与/LR 外/LL 商/RR 投/LL 资/RR 企/LL 业/RR 齐/LL 头/RR 并/LL 进/RR ，/LR 国/LL 有/RR 企/LL 业/RR 继/LL 续/RR 居/LL 于/RR 主/LL 导/RR 地/LL 位/RR ，/LR 外/LL 商/RR 投/LL 资/RR 企/LL 业/RR 仍/LL 然/RR 发/LL 挥/RR 重/LL 要/RR 的/LR 作/LL 用/RR 。/LR

(b) Segmented output

在｜刚刚｜过去｜的｜一九九七年｜，｜中国｜进出口｜贸易｜中｜，｜国有｜企业｜与｜外商｜投资｜企业｜齐头｜并进｜，｜国有｜企业｜继续｜居于｜主导｜地位｜，｜外商｜投资｜企业｜仍然｜发挥｜重要｜的｜作用｜。

(c) Gold Standard

在｜刚刚｜过去｜的｜一九九七年｜，｜中国｜进出口｜贸易｜中｜，｜国有｜企业｜与｜外商｜投资｜企业｜齐头并进｜，｜国有｜企业｜继续｜居于｜主导｜地位｜，｜外商｜投资｜企业｜仍然｜发挥｜重要｜的｜作用｜。

5. Results

In evaluating our model, we calculated both the tagging accuracy and segmentation accuracy. The calculation of the tagging accuracy is straightforward. It is simply the total number of correctly POC-tagged characters divided by the total number of characters. In evaluating segmentation accuracy, we used three measures: precision, recall and balanced F-score. Precision p is defined as the number of correctly segmented words divided by the total number of words in the automatically segmented corpus. Recall r is defined as the number of correctly segmented words divided by the total number of words in the gold standard, which is the manually annotated corpus. F-score f is defined as follows:

$$f = \frac{p \times r \times 2}{p + r} \quad (5)$$

The results of the three experiments are tabulated in Table 3:

Table 3. Experimental results

	Tagging accuracy		*Segmentation accuracy*			
Experiments	Training	Testing	Testing			
			p(%)	r(%)	f(%)	r(% new words)
1a	n/a	n/a	87.34	92.34	89.77	1.37
1b	n/a	n/a	94.51	95.80	95.15	n/a
2	97.90	96.05	95.01	94.94	94.98	70.20

The results from Experiment One show that the accuracy of the maximum matching algorithm degrades sharply when there are new words in the testing data, even when there is only a small proportion of them. Assuming an ideal scenario where there is no new word in the testing data, the maximum matching algorithm achieves an F-score of 95.15%. However, when there are new words (words not found the training data), the accuracy drops to only 89.77% in F-score. In contrast, the maximum entropy tagger achieves an accuracy of 94.98% by the balanced F-score even when there are new words in testing data. This result is only slightly lower than the 95.15% that the maximum matching algorithm achieves when there is no new word. An analysis of the new words (words not in the training data) is more revealing. Of the 510 words that are found in the testing data but not in the training data, 7 or 1.37% of them are correctly segmented by the maximum matching algorithm (Experiment 1a), while the maximum entropy model correctly segmented 70.20%, or 358 of them. The 7 words the maximum matching algorithm segmented correctly happen to be single-character words. This is expected because the maximum matching algorithm stops when it can no longer extend a string of *hanzi* based on a dictionary. In contrast, for the maximum entropy model, unknown words are predicted based on the distribution of their components. Even though the new words are not found in the training data, their components can still be found and words can be proposed based on the distribution of their components, a property that is typical of back-off statistical models. The fact the recall of the unknown words is well below the overall recall suggests that statistics of the unknown words are harder to collect than the known words.

The results of this segmenter against previous studies are harder to assess. One reason why this is difficult is that the accuracy representing segmenter performance can only be meaningfully interpreted if there is a widely accepted definition of wordhood in Chinese. It has been well-documented in the linguistics literature [Dai, 1992; Packard, 2000; Xue, 2001] that phonological, syntactic and semantic criteria do not converge to allow a single notion of "word" in Chinese. In practice, noting the difficulty in defining wordhood, researchers in

automatic word segmentation of Chinese text generally adopt their own working definitions of what a word is, or simply rely on native speakers' subjective judgments. The problem with native speakers' subjective judgements is that native speakers generally show great inconsistency in their judgments of wordhood, as should perhaps be expected given the difficulty of defining what a word is in Chinese. For example, Wu and Fung [1994] introduced an evaluation method which they call nk-blind. To deal with the inconsistency they proposed a scheme in which n human judges are asked to segment a text independently. They then compare the segmentation of an automatic segmenter with those of the human judges. For a given "word" produced by the automatic segmenter, there may be k human judges agreeing that this is a word, where k is between $zero$ and n. For eight human judges, the precision of the segmentation with which all the human judges agree is only 30%, while the precision of the segmentation that at least one human judge agrees with is 90%. [Sproat *et al*., 1996] adopted a different evaluation method since their work on Chinese word segmentation is tailored for use in a text-to-speech system. Their subjects, who have no training in linguistics, are instructed to segment sentences by marking all the places they might be plausibly pause if they were reading the text aloud. They tested inter-subject consistency on six native speakers of Mandarin Chinese and the average inter-subject consistency is 76%. These experiments attest the difficulty of evaluating the performance of different segmenters.

The situation is improving with the emergence of published segmentation standards and corpora manually segmented in keeping with these standards [Xia, 2000; Yu *et al*.,1998; CKIP, 1995]. Still, the corpora can vary by size, the complexity of the sentences in the corpora, so on and so forth. Unless the segmenters are tested with a single standard corpus, the performance of different segmenters are still hard to gauge. Still some preliminary observations can be made in this regard. Our accuracy is much higher that those reported in [Hockenmaier and Brew, 1998] and [Xue, 2001], who used error-driven transformation-based learning to learn a set of n-gram rules to do a series of merge and split operations on data from Xinhua news, the same data source as that of ours. The results they reported are 87.9% (trained on 100,000 words) and 90.2% (trained on 80,000 words) respectively, measured by the balanced F-score. Using a statistical model called prediction by partial matching (PPM), Teahan *et al*. [2000] reported a significantly better result. The model was trained on a million words from Guo Jin's Mandarin Chinese PH corpus and tested on five 500-segment files. The reported F-scores are in a range between 89.4% and 98.6%, averaging 94.4%. Since the data is also from Xinhua newswire, some comparison can be made between our results and this model. With less training data, our results using the maximum entropy model are slightly higher (by 0.48%). Tested on the same test data as ours, the Microsoft system [Wu, 2003] achieved a higher accuracy, achieving precision and recall rates of 95.98% and 96.36%

respectively, using a dictionary of around 89K words, compared with around 19K unique words in our training data. We believe our approach can achieve higher accuracy with more training data.

5.1 Personal names

It has long been noted that personal names often pose a serious problem for automatic word segmentation, presumably because new names are constantly made up and it is impossible to list them exhaustively in pre-compiled dictionaries that dictionary-based approaches heavily rely on. It is expected that these names should not generally be a problem for the present character-based approach in the same way because new words are not distinct problems for this approach. Among the 137 personal names (122 unique names, both Chinese names and foreign name transliterations) found in the testing data, 119 of them are segmented correctly, with a recall of 86.86%. The 18 wrongly segmented names are given in Table 4. In general, longer names, especially foreign names, are more likely to cause problems for this model.

Table 4. Incorrectly segmented personal names

Correct Segmentation	Segmenter Output
穆罕默德．胡期尼．穆巴拉克	穆罕｜默德．胡期尼．穆巴拉克
加央多吉	加央｜多｜吉
袁养和	袁养｜和
汪家(廖去广加金旁)	汪｜家｜(｜廖去｜广加金旁)
桑普拉斯	桑普拉斯伤愈
彭定康	彭定｜康道别
黄河明	黄河｜明以
顾明	顾明、
金硕仁	金硕｜仁
克里斯蒂娜．斯米贡	克里斯蒂娜．斯米贡。
江｜主席	江主席
米本育代	米｜本育代
中屋朱美	中屋｜朱美
里戈韦塔．门楚	家里戈韦塔．门楚
凯基特．荷布南南德	凯基特．荷布｜南南德
王咸儒	王咸儒说
里库佩罗	里库｜佩罗 23 日
令狐道成	令｜狐道成

5.2 Contribution of Features

In an effort to assess the effectiveness of the different types of features, we retrained our system by taking out each group of features in (5). The most effective features are the ones which, when not used, result in the most loss in accuracy. Table 5 shows that there is loss of accuracy when any of the six groups of features are not used. This means that all of the features in (5) made a positive contribution to the overall model. It is also clear that the features in (5d), which are pairs of Chinese characters, are the most effective. A substantial number of the features in (5d) encode two-character words and are thus good indicators of how the current character should be tagged. For example, if C_0 = 功 and C_1 = 能, this is good indication that 功 will start a word 功能, thus receive a tag LL. The previous (next) two characters individually (5c), the previous tags (5f) and the current character (5b) also made a substantial contribution to the overall model. The least useful features are the previous and the next character together (5e) and the default feature. The default feature is useful when no other features are invoked, e.g. when the current character is unknown and the previous two and next two characters are also unknown. It is not as effective as other features presumably because the likelihood of this scenario happening is small, given the characters in Chinese are limited in number.

Table 5. The effectiveness of different features

Features	*Tagging accuracy*	*Segmentation accuracy*		
		p(%)	r(%)	f(%)
all	96.05	95.01	94.94	94.98
w/o (a)	96.03	94.97	94.94	94.96
w/o (e)	95.92	94.85	94.86	94.85
w/o (b)	95.16	93.99	93.95	93.97
w/o (f)	95.41	93.88	93.95	93.91
w/o (c)	95.11	93.40	93.95	93.67
w/o (d)	92.62	91.04	91.06	91.05

5.3 Effects of Tag Sets

The choice of our POC tag set is based on linguistic intuitions. The use of four tags is linguistically intuitive in that LL tags morphemes that are prefixes or stems in the absence of prefixes, RR tags morphemes that are suffixes or stems in the absence of suffixes, MM tags stems with affixes and LR tags stems without affixes. The results in Table 6 show that our linguistically intuitive tag set is also the most effective. The use of three tags (LL for beginning of a word, RR for continuation of a word and LR for word by itself) that has been proven to be the most useful for baseNP chunking [Ramshaw and Marcus, 1995] results in comparable performance in segmentation accuracy. The use of two tags (LL for beginning of

a word and RR otherwise) results in substantial loss in segmentation accuracy while gaining in tagging accuracy. This is a somewhat surprising result since there is no inconsistent tagging with this tag set and thus no loss in accuracy in the post-tagging conversion process.

Table 6. The effectiveness of different tagsets

Tagset	*Tagging accuracy*	*Segmentation accuracy*		
		p(%)	r(%)	f(%)
Two	97.51	94.37	94.40	94.38
Three	96.51	95.09	94.83	94.96
Four	96.05	95.01	94.94	94.98

6. Conclusions and Future Work

The preliminary results show that the maximum entropy model can be effectively applied to Chinese word segmentation. It is more robust than the maximum matching algorithm in the sense that it can handle unknown words much more effectively. The results also show that our approach is competitive against other machine-learning models.

Much work needs to be done to evaluate this approach more thoroughly. For example, more experiments need to be performed on data sources other than the newswire type and on standards other than the Penn Chinese Treebank. In addition, we plan to explore ways to further improve this segmenter. For instance, we expect that the segmenter accuracy can still be improved as more training data become available. Refined pre-processing or post-processing steps could also help improve segmentation accuracy. For example, instead of tagging *hanzi* directly it might be possible to tag morphemes, which may or may not be composed of just one *hanzi*. There might also be better ways to convert a tagged sequence into a word sequence than the simple approach we adopted.

Acknowledgement

I would like to thank Susan Converse for her detailed comments on a previous version of this work. The comments of the three anonymous reviewers also led to significant improvement of this paper. I would also like to thank Andi Wu for for his help in evaluating the results reported here and Richard Sproat, the guest editor of this special issue, for help with references. All inadequacies that still exist in this work are my own, of course. This research was funded by DARPA N66001-00-1-8915.

References

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

CKIP, 1995. An Introduction to the Academia Sinica Balanced Corpus (in Chinese). Technical Report 95-02, Taipei: Academia Sinica.

Dai, Xiang-Ling, 1992. *Chinese Morphology and its Interface with the Syntax*. Ph.D. thesis, Ohio State University.

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(1):33–56.

Gan, Kok-Wee, 1995. *Integrating Word Boundary Disambiguation with Sentence Understanding* . Ph.D. thesis, National University of Singapore.

Gan, Kok-Wee, Martha Palmer, and Kim-Teng Lua, 1996. ''A statistically emergent approach for language processing: Application to modeling context effects in ambiguous Chinese word boundary perception". *Computational Linguistics*, 22(4):531–53.

Ge, Xianping, Wanda Pratt, and Padhraic Smyth, 1999. ''Dicovering Chinese words from unsegmented text". In *SIGIR'99*.

Gu, Ping and Yuhang Mao, 1994. ''Hanyu zidong fenci de jinlin pipei suanfa jiqi 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 Computating*. Singapore.

Guo, Jin, 1997. ''Critical tokenization and its properties". *Computational Linguistics*, 23(4):569–596.

Hockenmaier, Julia and Chris Brew, 1998. ''Error-driven segmentation of Chinese". *Communications of COLIPS*, 1(1):69–84.

Jin, Wangying and Lei Chen, 1998. ''Identifying unknown words in Chinese corpora". In *The First Workshop on Chinese Language Processing*. University of Pennsylvania, Philadelphia.

Liang, Nanyuan, 1993. ''shumian hanyu zidong fenci xitong cdws". *Journal of Chinese Information Processing*, 1(1):44–52.

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

Palmer, David, 1997. ''A trainable rule-based algorithm to word segmentation". In *Proceedings of the 35th Annual Meeting of the Association of Computational Linguistics*. Madrid, Spain.

Peng, Fuchun and Dale Schuurmans, 2001. "Self-supervised Chinese word segmentation". In F. Hoffman *et al* (ed.), *Advances in Intelligent Data Analysis, Proceedings of the Fourth International Conference (IDA-01)*. Heidelberg. Springer-Verlag.

Ramshaw, Lance and Mitchell P. Marcus, 1995. ''Text chunking using transformation-based learning". In *Proceedings of the Third ACL Workshop on Very Large Corpora* .

Ratnaparkhi, Adwait, 1996. ''A maximum entropy part-of-speech tagger". In *Proceedings of the Empirical Methods in Natural Language Processing Conference*. University of Pennsylvania.

Ratnaparkhi, Adwait, 1998. *Maximum Entropy Models for Natural Language Ambiguity Resolution*. Ph.D. thesis, University of Pennsylvania.

Sproat, R. and C. L. Shih, 1990. ''A statistical method for finding word boundaries in Chinese text". *Computer Processing of Chinese and Oriental Languages*, 4(4):336–351.

Sproat, R., Chilin Shih, William Gale, and Nancy Chang, 1996. ''A stochastic finite-state word-segmentation algorithm for Chinese". *Computational Linguistics*, 22(3):377–404.

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*.

Wu, Andi, this issue. ''Customizable segmentation of morphologically derived words in Chinese". *Computational Linguistics and Chinese Language Processing*.

Wu, Andi and Zixin Jiang, 1998. ''Word segmentation in sentence analysis". In *Proceedings of the 1998 International Conference on Chinese Information Processing*. Beijing, China.

Wu, Dekai and Pascale Fung, 1994. ''Improving Chinese tokenization with linguistic filters on statistical lexical acquisition". In *Proceedings of the Fourth Conference on Applied Natural Language Processing*.

Xia, Fei, 2000. The Segmentation Guidelines for Chinese Treebank Project. Technical Report IRCS 00-06, University of Pennsylvania.

Xia, Fei, Martha Palmer, Nianwen Xue, Mary Ellen Okurowski, John Kovarik, Fu-Dong Chiou, Shizhe Huang, Tony Kroch, and Mitch Marcus, 2000. "Developing Guidelines and Ensuring Consistency for Chinese Text Annotation". In *Proc. of the 2nd International Conference on Language Resources and Evaluation (LREC-2000)*. Athens, Greece.

Xue, Nianwen, 2001. *Defining and Automatically Identifying Words in Chinese*. Ph.D. thesis, University of Delaware.

Xue, Nianwen, Fu-Dong Chiou, and Martha Palmer, 2002. ''Building a large annotated Chinese corpus". In *The Proceedings of the 19th International Conference on Computational Linguistics*. Taipei, Taiwan.

Yu, Shiwen, Xuefeng Zhu, Hui Wang, and Yunyun Zhang, 1998. *The Grammatical Knowledge-base of Contemporary Chinese — A Complete Specification (in Chinese)*. Tsinghua University Press.

Computational Linguistics and Chinese Language Processing
Vol. 8, No. 1, February 2003, pp. 49-76

Measuring and Comparing the Productivity of Mandarin Chinese Suffixes

Eiji Nishimoto*

Abstract

The present study attempts to measure and compare the *morphological productivity* of five Mandarin Chinese suffixes: the verbal suffix *-hua*, the plural suffix *-men*, and the nominal suffixes *-r*, *-zi*, and *-tou*. These suffixes are predicted to differ in their *degree of productivity*: *-hua* and *-men* appear to be productive, being able to systematically form a word with a variety of *base* words, whereas *-zi* and *-tou* (and perhaps also *-r*) may be limited in productivity. Baayen [1989, 1992] proposes the use of corpus data in measuring productivity in word formation. Based on *word-token* frequencies in a large corpus of texts, his *token-based* measure of productivity expresses productivity as the probability that a new word form of an affix will be encountered in a corpus. We first use the token-based measure to examine the productivity of the Mandarin suffixes. The present study, then, proposes a *type-based* measure of productivity that employs the *deleted estimation* method [Jelinek & Mercer, 1985] in defining *unseen* words of a corpus and expresses productivity by the ratio of *unseen word types* to *all word types*. The proposed type-based measure yields the productivity ranking "*-men*, *-hua*, *-r*, *-zi*, *-tou*," where *-men* is the most productive and *-tou* is the least productive. The effects of corpus-data variability on a productivity measure are also examined. The proposed measure is found to obtain a consistent productivity ranking despite variability in corpus data.

Keywords: Mandarin Chinese word formation, Mandarin Chinese suffixes, morphological productivity, corpus-based productivity measure.

1. Introduction

1.1 Morphological Productivity

The focus of a study of *morphological productivity* is on *derivational affixation* that involves a *base* word and an affix [Aronoff, 1976], as seen in *sharp* + *-ness* → *sharpness*, *electric* + *-ity*

* Ph.D. Program in Linguistics, The Graduate Center, The City University of New York, 365 Fifth Avenue, New York, NY 10016, U.S.A.
e-mail: enishimoto@gc.cuny.edu

→ *electricity*, *child* + *-ish* → *childish*.[1] Native speakers of a language have intuitions about what are and are not acceptable words of their language, and if presented with non-existent, *potential* words [Aronoff, 1983], they accept certain word formations more readily than others [Anshen & Aronoff, 1981; Aronoff & Schvaneveldt, 1978; Cutler, 1980]. Most intriguing in the issue of productivity is that *the degree of productivity* varies among affixes, and many studies in the literature have been devoted to accounting for this particular aspect of productivity [see Bauer, 2001, and Plag, 1999, for an overview].

How the degree of productivity varies among affixes is best illustrated by the English nominal suffixes *-ness* and *-ity*, which are often considered "rivals" as they sometimes share a base word (e.g., *clear* → *clearness* or *clarity*). In general, -ness is felt to be more productive than *-ity*.[2] The word formation of -ity is limited, for example, by the *Latinate Restriction* [Aronoff, 1976: 51] that requires the base word to be of Latinate origin; hence, *purity* is acceptable but **cleanity* is not. In contrast, *-ness* freely attaches to a variety of base words of both Latinate and Germanic (native) origin; thus, both *pureness* and *cleanness* are acceptable. There are also some affixes that could be regarded as *unproductive*; for example, Aronoff and Anshen [1998: 243] note that the English nominal suffix *-th* (as in long → *length*) has long been unsuccessful in forming a new word that survives, despite attempts at terms like *coolth*. Varying degrees of productivity are also observed in Mandarin Chinese word formation. As will be discussed shortly, some Mandarin suffixes appear to be more productive than others.

1.2 Measuring the Degree of Productivity

Early studies on productivity mainly focused on restrictions on word formation and viewed the degree of productivity to be determined by such restrictions [Booij, 1977; Schultink, 1961; van Marle, 1985]. Booij [1977: 120], for example, considers the degree of productivity of a *word formation rule* to be inversely proportional to the amount of restrictions that the word formation rule is subject to. Although the view that productivity is affected by restrictions on word formation is certainly to the point, from a quantitative point of view, measuring productivity by the amount of restrictions on word formation is limited in that the restrictive weight of such restrictions is unknown [Baayen & Renouf, 1996: 87].

Baayen [1989, 1992] proposes a corpus-based approach to the quantitative study of productivity. His productivity measure uses word frequencies in a large corpus of texts to

---

[1] Excluded from the study of productivity are seemingly irregular word formations, or "oddities" [Aronoff, 1976: 20], such as *blendings* (e.g., *smoke* + *fog* → *smog*) and *acronyms* (e.g., *NATO*).

[2] *-ity* can be more productive than *-ness* depending on the type of base word; for instance, *-ity* is more productive than *-ness* when the base word ends with *-ile* as in *servile* [Aronoff, 1976: 36] or with *-ible* as in *reversible* [Anshen & Aronoff, 1981]. Still, overall, *-ness* is intuitively felt to be more productive than *-ity*.

express productivity as the probability that a new word form of an affix will be encountered in a corpus (see Section 3). Although Bauer [2001: 204] observes that a generally agreed measure of productivity is yet to be achieved in the literature, Baayen's corpus-based approach seems to be appealing and promising. Most importantly, since corpus data include productively formed words that are typically not found in a dictionary [Baayen & Renouf, 1996], corpus-based descriptions of productivity reflect how words are actually used.[3] The corpus-based approach is also timely, as linguists have growing interests in corpus data. The present study pursues the corpus-based approach to measuring productivity using a corpus of Chinese texts.

The outline of this paper is as follows. In Section 2, five Mandarin suffixes are introduced and are analyzed qualitatively based on observations in the literature. In Section 3, Baayen's *token-based* productivity measure is discussed, and the measure is applied to a corpus of Chinese texts to quantitatively analyze the productivity of the Mandarin suffixes. In Section 4, a *type-based* productivity measure is proposed, and its performance is evaluated. Also, some experiments are conducted to examine the effects of corpus-data variability on a productivity measure. Section 5 summarizes the findings.

2. Mandarin Chinese Suffixes

2.1 A Qualitative Analysis of Five Mandarin Suffixes

The present study examines the productivity of five Mandarin suffixes: the verbal suffix *-hua*, the plural suffix *-men*, and the nominal suffixes *-r*, *-zi*, and *-tou*.

The verbal suffix *-hua* 化 functions similarly to English *-ize* (and *-ify*):

(1) *xiàndài* 现代 'modern' → *xiàndàihuà* 现代化 'modernize'

Verbs formed with *-hua* can be used as nouns [Baxter & Sagart, 1998: 40], so *xiàndàihuà* 现代化 in (1) can also be interpreted as 'modernization'. Analogous to English *-ize* (and *-ify*), *-hua* systematically attaches to a variety of base words to form verbs, such as *gōngyèhuà* 工业化 'industrialize', *guójìhuà* 国际化 'internationalize', and *jìsuànjīhuà* 计算机化 'computerize'.

The suffix *-men* 们 pluralizes a noun, as in the following example:

(2) *xuésheng* 学生 'student' → *xuéshengmen* 学生们 'students'

According to Packard's [2000] classification, *-men* is a *grammatical affix*, whereas the other four suffixes that we examine are *word-forming affixes*. If we use the standard terminology of

[3] But see also Plag [1999] for a discussion of how dictionary data can be useful in a study of productivity.

the field, *-men* could be viewed as an *inflectional affix*, and the other four suffixes could be considered *derivational affixes*. There are three major characteristics of *-men* that differentiate *-men* from the English plural suffix *-s* [Lin, 2001: 59; Norman, 1988: 159; Ramsey, 1987: 64]. First, *-men* attaches only to human nouns[4]; hence, **zhuōzimen* 桌子们 'desks' and **diànnǎomen* 电脑们 'computers' are not acceptable, unless they are considered animate as in a cartoon. Second, *-men* is obligatory with pronouns (e.g., *wǒ* 我 'I' → *wǒmen* 我们 'we') but not with nouns; for example, *háizi* 孩子 without *-men* can be interpreted as 'child' or 'children' depending on the context. Third, *-men* is not compatible with numeral classifiers; hence, **sāngè xuéshengmen* 三个学生们 'three students' is ungrammatical. Due to these characteristics, *-men* may not be as frequently used or "productive" [Lin, 2001: 58] as the English plural suffix *-s*. However, *-men* has many base words to which it can attach, for there are a variety of nouns in Mandarin (as in any language) designating human beings (e.g., *jìzhěmen* 记者们 'reporters', *kèrénmen* 客人们 'guests', *shìzhǎngmen* 市长们 'mayors').

The suffix *-r* 儿 forms a noun from a verb or an adjective, or *-r* can create a diminutive form [Ramsey, 1987: 63; Lin, 2001: 57–58]:

(3) *huà* 画 'to paint' → *huàr* 画儿 'painting'

(4) *niǎo* 鸟 'bird' → *niǎor* 鸟儿 'small bird'

The use of *-r* is abundant in the colloquial speech of local Beijing residents, and three distinct usages of *-r* by local Beijing residents are identified [Chen, 1999: 39]. First, *-r* can create a semantic difference:

(5) *xìn* 信 'letter' → *xìnr* 信儿 'message'

Second, a form with *-r* may be habitually preferred to a form without it:

(6) *huā* 花 'flower' → *huār* 花儿 'flower'

Third, *-r* may be attached to a word solely for a stylistic reason. The use of *-r* in the last category is the most frequent among local Beijing residents [Chen, 1999: 39]. In both Mainland China and Taiwan, the use of *-r* is not favored especially in broadcasting, and *-r* words are rarely incorporated into the standard [Chen, 1999: 39; Ramsey, 1987: 64].

The suffixes *-zi* 子 and *-tou* 头 typically appear in the following constructions:

(7) **mào* 帽 → *màozi* 帽子 'hat'

(8) **mù* 木 → *mùtou* 木头 'wood'

In these examples, *-zi* and *-tou* combine with a *bound morpheme* that does not constitute a

[4] In colloquial speech, *-men* can occasionally attach to some animal nouns (e.g., *gǒurmen* 狗儿们 'doggies').

word by itself (i.e., neither **mào* 帽 nor **mù* 木 is a word).

Historically, the word formation of *-zi* and *-tou* appeared in the course of two changes in Chinese: a shift from monosyllabic to disyllabic words and a simplification of the phonological system [Packard, 2000: 265–266]. According to Packard [2000: 265], the shift toward disyllabic words occurred as early as in the Zhou dynasty (1000–700 BC) and underwent a large scale development during and after the Han dynasty (206 BC–AD 220). The phonological simplification, which occurred around the same time [Packard, 2000: 266], caused syllable-final consonants to be lost, and many single-syllable words that were once distinct became homophones [Li & Thompson, 1981: 44]. One possible account of how the two changes occurred is that the phonological simplification preceded as a natural linguistic process of phonetic attrition, and the shift toward disyllabic words occurred as a solution to the increase of homophonous syllables [Li & Thompson, 1981: 44; Packard, 2000: 266]. The increase of homophonous syllables was particularly significant in Mandarin [Li & Thompson, 1981: 44], and *-zi* and *-tou* played a role in the disyllabification of Mandarin words.

The word formation of *-zi* and *-tou* is not limited to bound morphemes [Lin, 2001: 58–59; Packard, 2000: 84]:

(9) *shū* 梳 'to comb' → *shūzi* 梳子 'comb'

(10) *xiǎng* 想 'to think' → *xiǎngtou* 想头 'thought'

In these examples, *-zi* and *-tou* form a noun by attaching to a free morpheme (i.e., both *shū* 梳 and *xiǎng* 想 are independent words).

The term "productive" is sometimes used in the literature to describe the above-discussed suffixes. Ramsey [1987: 63] describes *-tou* to be much less productive than *-zi*, while Li and Thompson [1981: 42–43] observe that *-zi* and *-tou* are both no longer productive. Lin [2001: 57] views *-r* to be the most productive Mandarin suffix. Unfortunately, the basis for these observations is left unclear. Some observations may be based on the number of word forms of a suffix found in a dictionary; for example, present-day Mandarin has by far more *-zi* word forms than *-tou* word forms, and this may lead to the view that *-zi* is more productive than *-tou*. However, as Aronoff [1980] argues, of interest to linguists is the *synchronic* aspect of productivity (i.e., how words of an affix can be formed at a given point in time), rather than the *diachronic* aspect of productivity (i.e., how many words of an affix have been formed between two points in time). Concentrating on the synchronic aspect, if we associate productivity with regularity in word formation [Spencer, 1991: 49] or availability of base words with which a new word can be readily formed, we may predict *-hua* and *-men* to be productive, and *-zi* and *-tou* to be limited in productivity. The productivity of *-r* would likely depend on the context—if we focus on broadcasting, the productivity of *-r* may also be limited. Admittedly, these predictions are speculative, and the difficulty in describing the productivity

of an affix is where a quantitative productivity measure becomes important. In the following sections, the productivity of the Mandarin suffixes will be examined quantitatively.

3. Quantitative Productivity Measurement

3.1 Baayen's Corpus-Based Approach

Baayen [1989, 1992] proposes a corpus-based measure of productivity, formulated as:

$$(11)\quad p = \frac{n_1}{N}$$

where given all word forms of an affix found in a large corpus of texts, n_1 is the number of word types of the affix that occur only once in the corpus, the so-called *hapax legomena* (henceforth, *hapaxes*), N is the sum of word tokens of the affix, and p is the productivity index of the affix in question.[5] The measure (11) employs Good's [1953] probability estimation method (commonly known as the *Good-Turing* estimation method) that provides a mathematically proven estimate [Church & Gale, 1991] of the probability of seeing a new word in a corpus, based on the probability of seeing hapaxes in that corpus. The productivity index p expresses the probability that a new word type of an affix will appear in a corpus after N tokens of the affix have been sampled. One important characteristic of the measure (11) is that it is *token-based*; that is, the measure relies on word-token frequencies in a corpus. The sum of word types of an affix in a corpus, represented by V, is not directly tied to the degree of productivity (see Section 4.1). In the remaining sections, the measure (11) will be referred to as the *hapax-based* productivity measure.[6]

While the hapax-based measure has been primarily used in the studies of Western languages, such as Dutch [e.g., Baayen, 1989, 1992] and English [e.g., Baayen & Lieber, 1991;

---

[5] A clear distinction has to be made between *word tokens* and *word types* in the context of a corpus study. To give the simplest example, if we have three occurrences of *the* in a small corpus, the token frequency of *the* is three, and the type frequency of *the* is one. In the case of affixation, we ignore the differences between singular and plural forms; for example, if we have a corpus that has {*activity*, *activity*, *activities*, *possibility*, *possibilities*}, the token frequency of *-ity* is five (the sum of all these occurrences of *-ity*) while the type frequency of *-ity* is two (after normalizing the plural forms, we have two distinct *-ity* words, *activity* and *possibility*). An exception to ignoring the plural suffix is when we are interested in the productivity of the plural suffix itself. In that case, if we have a corpus consisting of {*book*, *books*, *books*, *student*, *students*}, the token frequency of *-s* is three (i.e., *books*, *books*, and *students*), and the type frequency of *-s* is two (we have two distinct *-s* forms, *books* and *students*).

[6] For the purposes of this paper, the term *hapax-based measure* is used to express, in a shorthand manner, the fact that the measure defines new words based on hapaxes and that the measure is token-frequency-based. It should not be confused with the *hapax-conditioned measure*, p^*, discussed in Baayen [1993].

Baayen & Renouf, 1996], the measure was also used by Sproat and Shih [1996] in a study of Mandarin word formation. The focus of Sproat and Shih's study was on productivity in Mandarin *root compounding*, as seen in the nominal root *yǐ* 蚁 of *mǎyǐ* 蚂蚁 'ant' that forms many words of 'ant-kind', such as *yǐwáng* 蚁王 'queen ant' and *gōngyǐ* 工蚁 'worker ant'. By analyzing the degree of productivity of a number of Mandarin nominal roots, Sproat and Shih showed that, contrary to a claim in the literature, root compounding is a productive word-formation process in Mandarin. For example, while *shí* 石 'rock-kind' and *yǐ* 蚁 'ant-kind' had the productivity indices of 0.129 and 0.065, respectively, apparently unproductive *bīn* 槟 and *láng* 榔 of *bīnláng* 槟榔 'betel nut' were found to have zero productivity. Sproat and Shih's study shows that a corpus-based study of productivity in Chinese is fruitful.

3.2 A Corpus of Segmented Chinese Texts

A major difficulty in conducting a corpus-based study of productivity in Chinese is that Chinese texts lack word delimiters. Segmentation of Chinese text is, by itself, a contested subject [see Sproat, Shih, Gale, & Chang, 1996], and consequently, a large-size corpus of segmented Chinese texts is not as readily available as a large-size corpus of English texts. Sproat and Shih [1996] used a large-size Chinese corpus (40-million Chinese characters) in their study by running an automatic segmenter to segment strings that contained the Chinese characters of interest and manually processing some problematic cases where the segmentation was not complete.

The corpus of choice in the present study is a "cleaned-up" version of *the Mandarin Chinese PH Corpus* [Guo, 1993; hereafter, *the PH Corpus*] of segmented Chinese texts, made available in a study by Hockenmaier and Brew [1998].[7] The corpus contains about 2.4-million (2,447,719) words—or 3.7-million (3,753,291) Chinese characters—from *XinHua* newspaper articles between January 1990 and March 1991. The texts of the PH Corpus are originally encoded in *GB* (simplified Chinese characters), and to facilitate the processing of the texts in computer programs, we convert the texts into *UTF8* (*Unicode*) using an encoding conversion program developed by Basis Technology [Uniconv, 1999]. The size of the PH Corpus is relatively small by today's standards (cf. a corpus of 80-million English words used in Baayen & Renouf, 1996), but the PH Corpus is one of few widely available corpora of segmented Chinese texts. Another widely available corpus of segmented Chinese texts is *the Academia Sinica Balanced Corpus* [1998; hereafter, *the Sinica Corpus*] that contains 5-million words from a variety of text sources. The sentences of the Sinica Corpus are syntactically parsed, so the *part-of-speech* of each segmented word is identified. Although the Sinica Corpus is not

[7] The PH Corpus can be downloaded from the ftp server of the Centre for Cognitive Science at University of Edinburgh.

used in the present study, the use of the Sinica Corpus is certainly of interest.[8]

Certain words were filtered out as potentially relevant words of the Mandarin suffixes in question were collected from the PH Corpus. With *-r* and *-zi*, a criterion for distinguishing a suffix from a non-suffix is that *-r* and *-zi* as a suffix lose their tone [Liu, 2001, 57–58; Norman, 1988, 113–114]. This criterion helps identify and block many non-suffixal cases where *-r* and *-zi* denote 'son' or 'child', such as *yīng'ér* 婴儿 'baby', *fùzǐ* 父子 'father and son', and *xiàozǐ* 孝子 'filial son'.[9] We exclude *wénhuà* 文化 'culture' because it is never a verb, and according to Norman [1988: 21], the specific use of *wénhuà* 文化 to mean 'culture' was adopted from Japanese. Also excluded are some *-tou* words, such as *máotóu* 矛头 'spearhead', in which *-tou* is a bound morpheme denoting 'head'. In addition, all pronouns in *-men* are excluded, as suggested in Sproat [2002]. As discussed earlier, *-men* behaves differently between pronouns and nouns (i.e., it is obligatory only with pronouns), and it is *-men* attaching to open-class nouns, rather than closed-class pronouns, that we are currently interested in.

3.3 A Quantitative Analysis of the Mandarin Suffixes

The result of the hapax-based measure applied to the PH Corpus is shown in Table 1. Figure 1 presents a bar graph illustrating the productivity ranking of the suffixes based on the *p* values.

Table 1. The result of the hapax-based productivity measure applied to the PH Corpus

suffix	*V*	*N*	n_1	*p*
-r	35	184	14	0.076
-men	219	2324	101	0.043
-zi	177	2130	62	0.029
-hua	209	3366	93	0.028
-tou	36	600	6	0.010

Note. With all the occurrences of a suffix found in the corpus, *V* is the sum of types, *N* is the sum of tokens, n_1 is the number of hapaxes, and *p* is the productivity index of the suffix. The suffixes are sorted in descending order by *p*.

[8] The use of the PH Corpus in the present study is solely due to the fact that the computer programs currently used were written for the PH Corpus. It must be noted, however, that findings from a larger, more balanced corpus do not necessarily minimize findings from a smaller, less balanced corpus. Findings from both the PH Corpus (a small corpus of newspaper texts) and the Sinica Corpus (a large corpus of a variety of texts) are of interest because corpora of different types enable a comparison of findings by the corpus type.

[9] Note in these examples that the tone of *-r* and *-zi* is retained (i.e., *-ér* and *-zǐ*, respectively). *-r* is originally *-ér*, and it becomes *-r* as a suffix, as a result of losing its syllabicity [Norman, 1988: 114].

Among the five suffixes, *-r* is found to be the most productive. The high productivity of *-r* is somewhat unexpected given the fact that the PH Corpus consists of newspaper texts. If the use of *-r* is not favored in broadcasting, we may also expect a limited use of *-r* in a newspaper context. In addition, the use of *-r* is often a mere phonological phenomenon as seen in the speech of local Beijing residents, and it is unlikely for such a phonological phenomenon to be represented in newspaper texts. In Table 1, the number of types (*V*) of *-r* does not reach the number of types of the least productive suffix *-tou*. However, the token frequency (*N*) of *-r* is lower than that of *-tou*, and *-r* has a larger number of hapaxes than *-tou*. Under the hapax-based measure, a high token frequency is associated with a high *degree of lexicalization of words* (i.e., the extent to which words are stored in the lexicon in their full form), and a high degree of lexicalization of words, in turn, is associated with a low degree of productivity [Baayen, 1989, 1992]. The rationale behind this mechanism is that if many words of an affix are lexicalized, the word formation rule of the affix needs to be invoked less often to form a word. What the present data of *-r* indicate, then, is that *-r* words are characterized by a low degree of lexicalization. The low degree of lexicalization of *-r* words and the relatively large number of hapaxes (as compared with *-tou*) suggest that the word formation rule of *-r* is active.

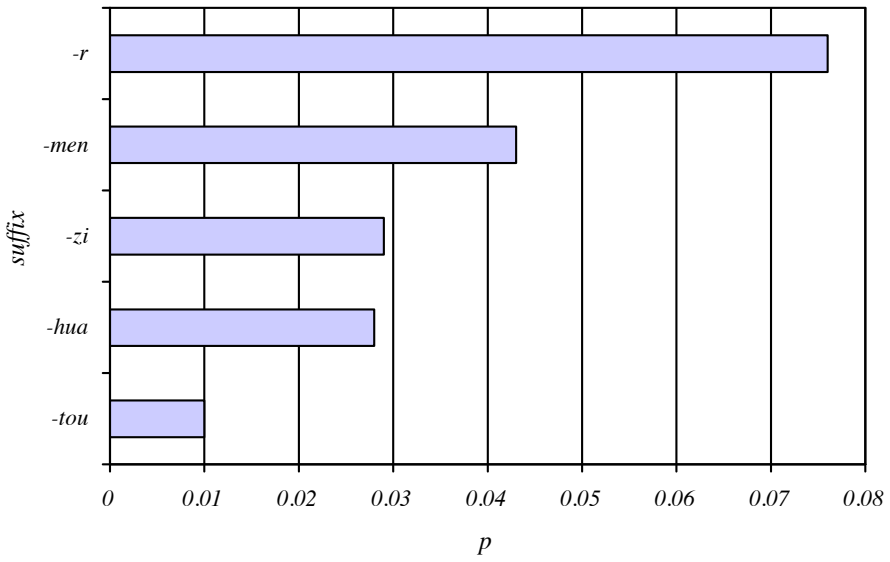

Figure 1 The productivity ranking of the Mandarin suffixes by the p values (the vertical axis lists the suffixes, and the horizontal axis shows the p values of the suffixes).

The productivity of *-hua* seems somewhat lower than what we may expect from the regularity in *-hua* word formation. Comparing *-men* and *-hua* in Table 1, we see that *-men* and *-hua* are similar with respect to both V and n_1, but the p value of *-hua* is lowered by the high token frequency (N) of *-hua*. The high token frequency of *-hua* could be attributed to the fact that the present analysis includes *-hua* words used as nouns. According to Baxter and Sagart [1998: 40], *-hua* words are formed as verbs first, and these verbs can be used as nouns. If this is the case, the word formation of *-hua* is also relevant in *-hua* nouns. However, the uniform treatment of *-hua* verbs and *-hua* nouns may not be appropriate for the hapax-based measure. It could be the case, for example, that some *-hua* words are typically used as nouns with high token frequencies while other *-hua* words are typically used as verbs with low token frequencies. It is, therefore, necessary to make a more detailed analysis of the word frequency distribution of *-hua* by separating *-hua* nouns from *-hua* verbs. Distinguishing nouns from verbs is unfortunately not available in the PH Corpus due to lack of syntactic information. A clearer description of the productivity of *-hua* could be achieved with a syntactically parsed corpus such as the Sinica Corpus.

4. Type-Based Deleted Estimation

4.1 Type-Based Measures

The present study explores a *type-based* measure of productivity. It has been argued that the sum of types of an affix in a corpus, V, alone often leads to some unintuitive results in measuring productivity [Baayen, 1989, 1992; Baayen & Lieber, 1991].[10] For example, Baayen and Lieber [1991: 804] point out that the type frequencies of *-ness* and *-ity* in their corpus (497 and 405, respectively) do not adequately represent the fact that *-ness* is intuitively felt to be much more productive than *-ity*. If the number of types in a corpus can be misleading with respect the degree of productivity, how can we make use of type frequencies in a productivity measure?

An early attempt at a type-based measure of productivity was made by Aronoff [1976: 36], in which he proposed that the degree of productivity of an affix could be measured by the ratio of the number of actual words of the affix to the number of *possible words* of the affix. The measure is described by Baayen [1989: 28] as:

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

where V is the number of actual words with the relevant affix, S is the number of possible words with the affix, and I is the productivity index of the affix. Baayen [1989: 28] argues that

[10] See Baayen [1992] and Baayen and Lieber [1991] for a discussion of the *global productivity* of an affix (expressed as P^*) based on a two-dimensional analysis of p and V.

the measure lacks specification on how to obtain V and S. Moreover, he argues that the measure can be interpreted to express, ironically, the degree of "unproductivity" of an affix because the number of possible words (S) would be, in theory, increasingly large (hence, the productivity index I would be increasingly small) for a very productive affix [Baayen, 1989: 30].

Baayen [1989, 1992] defines V and S based on corpus data. V is (as before) the sum of types of the relevant affix found in a corpus, and S (expressed as $\hat{S}$) is statistically estimated for an infinitely large corpus; that is, $\hat{S}$ is the number of possible word types of the relevant affix to be expected when the corpus size is increased infinitely.[11] The measure that Baayen [1989: 60] proposes:

(13) $$I = \frac{\hat{S}}{V}$$

is the inverse of (12) and expresses the *potentiality of word formation rules*, the extent to which the number of actual word types of an affix exhaust the number of possible word types of the affix [Baayen, 1992: 122]. The measure (13), however, is not considered an alternative measure of the degree of productivity [Baayen, 1992: 122].

What does not appear to have been explored so far is the question of what *new words* would mean under a type-based measure. One major appeal of the hapax-based measure is that it centers on the formation of new words, and we may wish to try focusing on the formation of new words under a type-based measure. However, a problem with taking a type-based approach is that we can no longer rely on the Good-Turing estimation method. In the next section, we will discuss another method of defining new words of a corpus.

4.2 The Deleted Estimation Method

To define new words of a corpus in a type-based manner, we can employ the *deleted estimation* method [Jelinek & Mercer, 1985] used in language engineering. In a probabilistic language model, given a training corpus and a test corpus, we process words in the test corpus based on the probabilities of word occurrence in the training corpus. Since not all words of the test corpus appear in the training corpus, we need a method of assigning an appropriate probability mass to the *unseen words* in the test corpus. The main task involved here is to adjust the probabilities of word occurrence in the training corpus so that non-zero probability can be assigned to unseen words of the test corpus. A method used in this probability adjustment, if incorporated into a productivity measure, can tell us the probability of encountering unseen words in a corpus. The Good-Turing estimation method underlying the

[11] The statistical techniques for obtaining $\hat{S}$, which involve an extended version of Zipf's law, are beyond the scope of this paper. For more details, the reader is referred to Baayen [1989, 1992].

hapax-based measure is widely used in probabilistic language modeling, and its successful performances are reported in the literature [Chen & Goodman, 1998; Church & Gale, 1991]. While the Good-Turing estimation method is a *mathematical* solution to the task of probability adjustment, where the needed probability adjustment is mathematically determined, the deleted estimation method is an *empirical* solution, where the needed adjustment is determined by comparing discrepancies in word frequency between corpora [Church & Gale, 1991; Manning & Schütze, 1999].

The deleted estimation method, when incorporated into a type-based productivity measure, proceeds as follows. We begin by preparing two corpora of the same size and text type. The easiest way to have two such corpora is to split a large corpus in the middle into two sub-corpora, which we will call *Corpus A* and *Corpus B*.[12] Comparing word types that appear in Corpus A against word types in Corpus B, *unseen word types* (or *unseen types*) in Corpus A are defined as those word types that do not appear in Corpus B. Likewise, unseen types in Corpus B are those that are absent in Corpus A. We obtain the average number of unseen types between Corpus A and Corpus B. Defining *all word types* (or *all types*) in a corpus as all the word types found in that corpus,[13] we also obtain the average number of all types between the two sub-corpora. The ratio of the average number of unseen types to the average number of all types expresses the extent to which word types of an affix are of an unseen type. With an assumption that unseen types are good candidates for new word types, the degree of productivity expressed in this manner comes close to Anshen and Aronoff's [1988: 643] definition of productivity as "the likelihood that new forms will enter the language."

The type-based deleted estimation productivity measure is formulated as follows:

Given Corpus A and Corpus B of the same size and text type, and all word types of an affix found in these corpora,

$$(14)\quad P_{tde}(A,B) = \frac{"\textit{unseen types in A given B}" + "\textit{unseen types in B given A}"}{"\textit{all types in A}" + "\textit{all types in B}"}$$

where *all types* of a corpus are all the word types found in that corpus, *unseen types* in one corpus are those that are absent in the other corpus, and P_{tde} is the degree of productivity of the affix in question (*tde = type-based deleted estimation*). In calculating P_{tde} by the measure (14), we can first average the unseen types in the nominator and the all types in the denominator. This will conveniently give us the average number of unseen types and the average number of all types, which are both of interest by themselves, before examining the ratio of the two (as

[12] These sub-corpora would be labeled *retained* and *deleted* (hence the term *deleted estimation*) under the original deleted estimation method. However, in the present context, we can simplify the argument by using the labels *Corpus A* and *Corpus B*.

[13] The number of *all types* is essentially the same as *V*.

will be seen later in Table 2). In the remaining sections, the measure (14) will be referred to as the P_{tde} measure. Using a Venn Diagram, Figure 2 illustrates elements involved in the P_{tde} measure.

Given $\mathbf{A} = \{a_1, ..., a_m\}$ *from Corpus A, and* $\mathbf{B} = \{b_1, ..., b_n\}$ *from Corpus B, where* a_i *and* b_i *are word types of an affix found in the two corpora,*

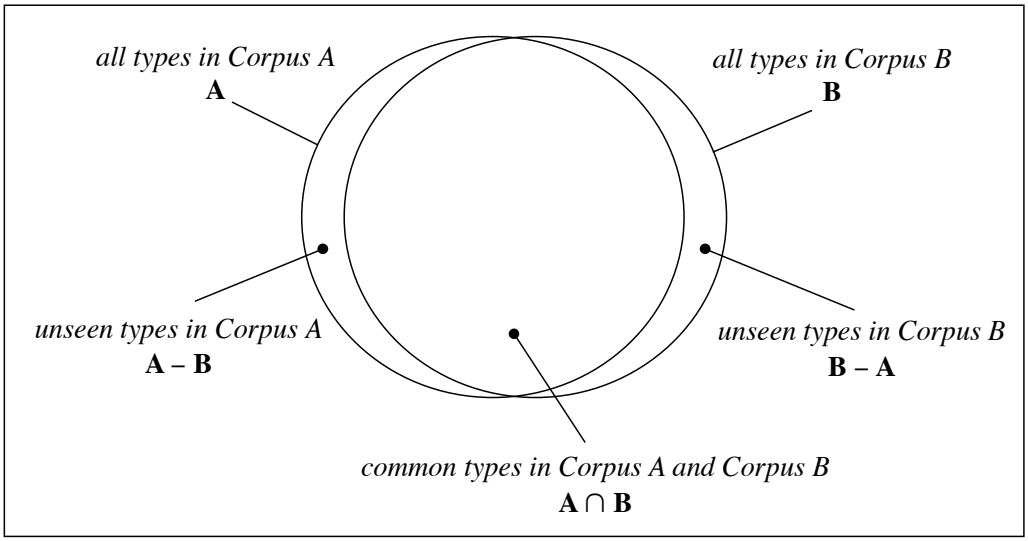

Figure 2 An illustration of elements involved in the P_{tde} measure (all types in a corpus are all the word types found in that corpus, unseen types in one corpus are those that are absent in the other corpus, and common types are the word types shared by the two corpora).

As a byproduct, the P_{tde} measure also identifies *common types*, word types that are shared by two sub-corpora, as shown in Figure 2. One possible interpretation of these common types is that they represent attested words, where attested words are defined as those words that are familiar to the majority of speakers. Although an approximation,[14] common types may be good candidates for attested words because unseen types, which are less likely to be familiar to the majority of speakers, are maximally excluded. As the corpus size increases, the number of common types may begin to provide a good estimate of the range of word types that are

[14] Strictly speaking, any word type with the token frequency of two or more in the original whole corpus has a chance to be shared by the two sub-corpora after the corpus is split. Thus, a word that appears only twice in a large corpus could be identified as a common type.

shared by the majority of speakers. Such a range of word types differs from the range of word types in a dictionary. Common types will not be pursued in the present study, but they may be worth further investigation in future research.

4.3 Performance of the P_{tde} Measure

The result of the P_{tde} measure applied to the PH Corpus is shown in Table 2. Figure 3 presents a bar graph that illustrates the productivity ranking of the suffixes based on the P_{tde} values.

Table 2. The result of the P_{tde} measure applied to the PH Corpus

suffix	*(average) all types*	*(average) unseen types*	P_{tde}
-men	149	70	0.470
-hua	144	65	0.451
-r	24.5	10.5	0.429
-zi	130.5	46.5	0.356
-tou	29.5	6.5	0.220

Note. The PH Corpus is split in the middle into two sub-corpora. *All types* in a sub-corpus are all the word types that appear in that sub-corpus. The second column shows the average number of all types between the two sub-corpora. *Unseen types* are those that appear in one sub-corpus but are absent in the other sub-corpus. The third column shows the average number of unseen types between the two sub-corpora. The tenths place in the second and third columns is due to the averaging. P_{tde} is the ratio of (*average*) *unseen types* to (*average*) *all types*. The suffixes are sorted in descending order by P_{tde}.

In Table 2, we find that *-r* is not as highly productive as under the hapax-based measure, though it still appears to be grouped with the more productive suffixes. Here, we may wonder why we examine the ratio of unseen types to all types, instead of examining the number of unseen types only. If productivity is determined by the number of unseen types only, *-r* would be among the less productive suffixes. However, comparing the number of unseen types alone is not satisfactory because an affix with a low frequency of use would generally be found to be less productive. The P_{tde} measure must be able to capture the possibility that an affix with a low frequency of use can nevertheless be productive when it is used to form a word. With respect to the present data, the ratio of unseen types to all types is relatively high for *-r*, indicating that a large proportion of *-r* word types are of an unseen type, or a potentially new type.

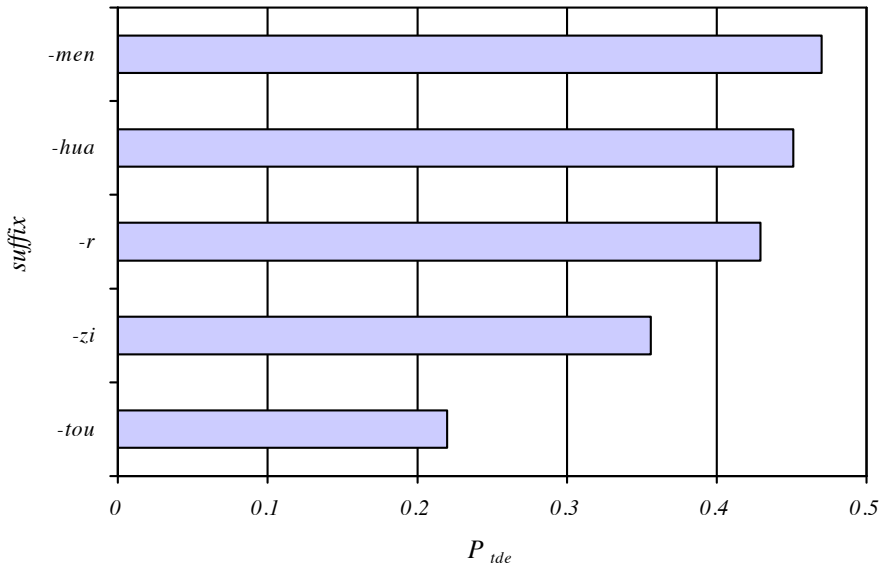

Figure 3 The productivity ranking of the Mandarin suffixes by the P_{tde} values (the vertical axis lists the suffixes, and the horizontal axis shows the P_{tde} values of the suffixes).

As was the case under the hapax-based measure, *-men* is found to be highly productive and *-tou* is found to be the least productive. The uniform treatment of *-hua* verbs and *-hua* nouns does not seem to pose a problem, though it is also of interest to investigate the effect of separating *-hua* nouns from *-hua* verbs under the P_{tde} measure.

The P_{tde} measure defines unseen types irrespective of word-token frequencies; that is, an unseen type in a corpus is "unseen" as long as it is absent in the other corpus, regardless of how many times the word is repeated in the same corpus. Figure 4 shows the word-token frequency distribution of unseen types in Corpus A and Corpus B. The labels used for the word-token frequency categories are: n_1 = words occurring once, n_2 = words occurring twice, ..., n_{5+} = words occurring five times or more.

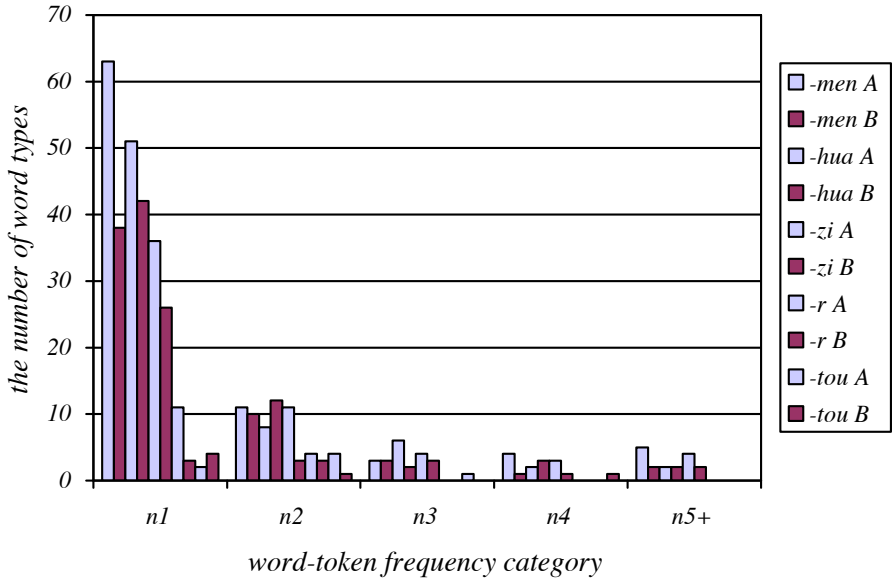

Figure 4 The word-token frequency distribution of unseen types in the two sub-corpora of the PH Corpus, Corpus A and Corpus B (the horizontal axis shows the word-token frequency category, and the vertical axis shows the number of word types in each frequency category; the letter following each suffix in the legend indicates from which sub-corpus the data are drawn; the order of the suffixes in the legend (from top down) corresponds to the order of bars in each frequency category (from left to right)).

We find in Figure 4 that the majority of unseen types are hapaxes. There are, nonetheless, unseen types that appear more than once in a corpus—some unseen types appear even five times or more (n_{5+}). We also notice gaps between the two sub-corpora in the word frequency of the unseen types (e.g., compare the number of *-men* hapaxes). Variability between two corpora will be the topic of discussion in the next section.

4.4 Variability in Corpus Data

Under the P_{tde} measure, a corpus is split in the middle to create two sub-corpora. So far, we have made the assumption that splitting a corpus in the middle would create two sub-corpora that are similar with respect to the text type. However, we must be cautious about this assumption. Baayen [2001] discusses how the texts and word frequency distribution of a

corpus can be non-uniform.[15] One way to reduce variability between split halves of a corpus is to randomize words of the corpus before splitting the corpus into two. Randomization of words can be accomplished by shuffling words; that is, given a corpus of n words, we exchange each i-th word (i = 1, 2, ..., n) with a randomly chosen j-th word ($1 \leq j \leq n$). If we repeat the "random split" of a corpus (i.e., randomizing words of a corpus and splitting the corpus in the middle) for a large number of times, say 1,000 times, and compute the mean of the relevant data, we should be able to obtain a stable, representative result of a productivity measure.[16] Table 3 shows the result of the hapax-based measure applied to the two sub-corpora of the PH Corpus, with and without randomization of words.

Table 3. The result of the hapax-based productivity measure applied to the two sub-corpora of the PH Corpus, Corpus A and Corpus B, with and without randomization of words

(a) Without randomization, a single split									
Corpus A					*Corpus B*				
suffix	*V*	*N*	n_1	*p*	*suffix*	*V*	*N*	n_1	*p*
-r	29	113	13	0.115	*-r*	20	71	6	0.085
-men	165	1183	84	0.071	*-zi*	119	841	53	0.063
-hua	148	1599	72	0.045	*-men*	133	1141	60	0.053
-zi	142	1289	57	0.044	*-tou*	29	256	8	0.031
-tou	30	344	5	0.015	*-hua*	140	1767	55	0.031
(b) With randomization, the mean of 1000 splits									
Corpus A					Corpus B				
suffix	*V*	*N*	n_1	*p*	*suffix*	*V*	*N*	n_1	*p*
-r	26	93	12	0.133	*-r*	26	91	12	0.130
-men	158	1164	77	0.067	*-men*	157	1160	77	0.066
-zi	138	1075	54	0.050	*-zi*	137	1055	54	0.051
-hua	154	1680	71	0.042	*-hua*	152	1686	69	0.041
-tou	31	303	8	0.025	*-tou*	31	297	8	0.027

Note. Each value in Part (b) is the mean of 1,000 random splits. The suffixes in each section are sorted in descending order by *p*. In Corpus B of Part (a), the *p* values of *-tou* and *-hua* expressed to the fourth decimal place are 0.0313 and 0.0311, respectively.

[15] See Baayen [2001] for an in-depth discussion of techniques for measuring variances among segments of a corpus.

[16] The procedure described here is thanks to suggestions by Baayen [personal communication].

In Part (a) of Table 3, the difference in V between Corpus A and Corpus B is almost significant,[17] which suggests variability in texts between the two sub-corpora, and a different productivity ranking is obtained in each sub-corpus. However, if we turn to Part (b) of Table 3, the productivity ranking becomes consistent between the two sub-corpora.[18] Interestingly, the productivity ranking in Part (b) of Table 3 is the same as one obtained earlier in Table 1 in Section 3.3. The p values in Part (b) of Table 3 are overall higher than those in Table 1, but this is an expected outcome, for p is dependent on the size of a corpus [Baayen, 1989, 1992; Baayen & Lieber, 1991]. We find that the hapax-based measure can achieve stability by means of a large number of random splits of a corpus.

What will be the effects of corpus-data variability on the P_{tde} measure? To examine this, we need to temporarily simplify the P_{tde} measure so that the value of P_{tde} will be obtained for each individual sub-corpus (without averaging unseen types and all types between two sub-corpora). That is, under the simplified measure, P_{tde} for Corpus A, P_{tde}(A), will be the ratio of "unseen types in A given B" to "all types in A"; and similarly, P_{tde}(B) will be the ratio of "unseen types in B given A" to "all types in B." Table 4 shows the result of the simplified P_{tde} measure applied to the two sub-corpora of the PH Corpus, with and without randomization of words.

The simplified P_{tde} measure is found to be quite vulnerable to corpus-data variability. In Part (a) of Table 4, the difference between Corpus A and Corpus B is almost significant in *all types* and *unseen types*, and the P_{tde} values differ significantly between the two sub-corpora.[19] However, if we turn to Part (b) of Table 4, the productivity ranking becomes consistent between the two sub-corpora.[20] Similarly to the hapax-based measure, the P_{tde} measure can achieve stability through a large number of random splits of a corpus.

---

[17] A paired t-test reveals that the difference in V approaches significance [$t(4) = 2.595, p = .06$], though the difference is not significant in other elements: N[$t(4) = .905, p > .10$], n_1[$t(4) = 2.046, p > .10$], and p [$t(4) = .555, p > .10$].

[18] The correlation coefficient between Corpus A and Corpus B improves in p after the random splits: p [$r(5) = (.850 \rightarrow) 1.0, p < .01$].

[19] A paired t-test shows that the difference approaches significance in *all types* [$t(4) = 2.595, p = .06$] and in *unseen types* [$t(4) = 2.595, p = .06$] and the difference is significant in P_{tde} [$t(4) = 2.869, p < .05$].

[20] The correlation coefficient between Corpus A and Corpus B improves in P_{tde} after the random splits: P_{tde} [$r(5) = (.753 \rightarrow) 9.99, p < .01$].

Table 4. The result of the simplified P_{tde} measure applied to the two sub-corpora of the PH Corpus, Corpus A and Corpus B, with and without randomization of words

(a) Without randomization, a single split							
Corpus A				*Corpus B*			
suffix	*all*	*unseen*	P_{tde}	*suffix*	*all*	*unseen*	P_{tde}
-men	165	86	0.521	*-hua*	140	61	0.436
-r	29	15	0.517	*-men*	133	54	0.406
-hua	148	69	0.466	*-r*	20	6	0.300
-zi	142	58	0.408	*-zi*	119	35	0.294
-tou	30	7	0.233	*-tou*	29	6	0.207
(b) With randomization, the mean of 1000 splits							
Corpus A				*Corpus B*			
suffix	*all*	*unseen*	P_{tde}	*suffix*	*all*	*unseen*	P_{tde}
-men	158	62	0.394	*-men*	157	61	0.389
-hua	154	57	0.372	*-hua*	152	55	0.364
-r	26	9	0.356	*-r*	26	9	0.342
-zi	138	40	0.291	*-zi*	137	39	0.287
-tou	31	5	0.160	*-tou*	31	5	0.163

Note. Each value in Part (b) is the mean of 1,000 random splits. The suffixes in each section are sorted in descending order by P_{tde}.

Figure 5 shows the word-token frequency distribution of unseen types averaged over the 1,000 random splits. We see in Figure 5 that unseen types with higher token frequencies (e.g., n_4 and n_{5+}) are almost absent. What this indicates is that as a result of randomizing words of a corpus, it became unlikely for unseen types to include word types that are repeated many times in a corpus. As compared with what we saw earlier in Figure 4, the greater majority of unseen types are now hapaxes, and variances between Corpus A and Corpus B are also reduced.

We now consider the P_{tde} measure in its original state (as in Section 4.2, with the averaging of unseen types and all types between two sub-corpora). Comparing Table 2 and Part (b) of Table 4, we find that the original P_{tde} measure achieves a result that is highly correlated with the result obtained with the 1,000 random splits.[21] Note in particular that the

[21] Comparing the elements of Table 2 and the elements of Corpus A in Part (b) of Table 4, the correlation coefficient is significant in all elements: *all types* [$r(5) = 1.0, p < .01$], *unseen types* [$r(5) = 1.0, p < .01$], and P_{tde} [$r(5) = 1.0, p < .01$]. Likewise, the correlation coefficient is significant in all elements when we compare the elements of Table 2 and the elements of Corpus B in Part (b) of Table 4: *all types* [$r(5) = 1.0, p < .01$], *unseen types* [$r(5) = 1.0, p < .01$], and P_{tde} [$r(5) = .999, p < .01$].

productivity ranking is consistent between Table 2 and Part (b) of Table 4. The P_{tde} measure seems to reduce the effects of corpus-data variability by averaging unseen types and all types between two sub-corpora. This is an advantage and makes the P_{tde} measure handy, for a large number of random splits of a corpus can be computationally expensive, especially when the corpus size is large.

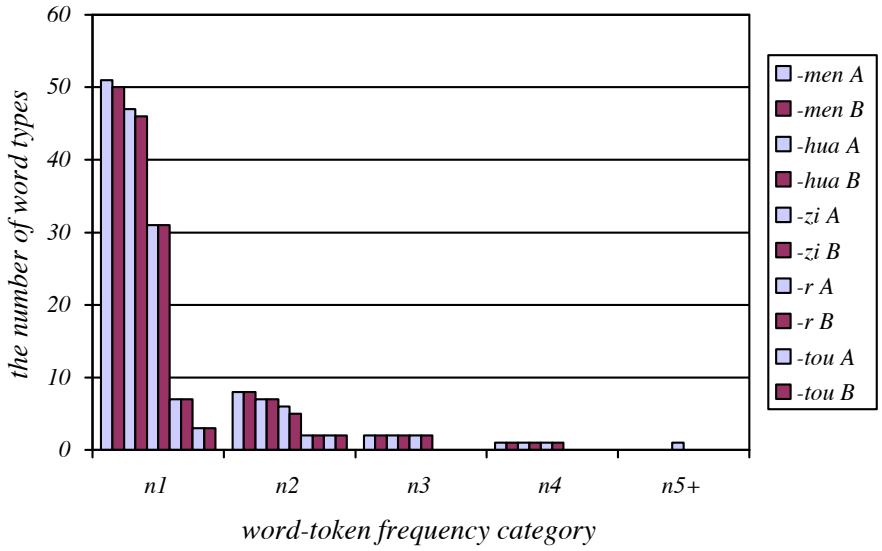

Figure 5. The word-token frequency distribution of unseen types in the two sub-corpora of the PH Corpus, Corpus A and Corpus B, averaged over 1000 random splits (the horizontal axis shows the word-token frequency category, and the vertical axis shows the number of word types in each frequency category; the letter following each suffix in the legend indicates from which sub-corpus the data are drawn; the order of the suffixes in the legend (from top down) corresponds to the order of bars in each frequency category (from left to right)).

5. Conclusion

The present study has proposed a type-based measure of productivity, the P_{tde} measure, that uses the deleted estimation method [Jelinek & Mercer, 1985] in defining unseen word types of a corpus. The measure expresses the degree of productivity of an affix by the ratio of unseen word types of the affix to all word types of the affix. If the ratio is high for an affix, a large proportion of the word types of the affix are of an unseen type, indicating that the affix has a great potential to form a new word.

We have tested the performance of the P_{tde} measure as well as the hapax-based measure of Baayen [1989, 1992] in a quantitative analysis of the productivity of five Mandarin suffixes: *-hua*, *-men*, *-r*, *-zi*, and *-tou*. The P_{tde} measure describes *-hua*, *-men*, and *-r* to be highly productive, *-zi* to be less productive than these three suffixes, and *-tou* to be the least productive, yielding the productivity ranking "*-men*, *-hua*, *-r*, *-zi*, *-tou*." The P_{tde} measure and the hapax-based measure rank the suffixes differently with respect to *-hua* and *-r*. The relatively low productivity of *-hua* under the hapax-based measure could be attributed to the inclusion of *-hua* nouns in the present analysis. *-r* is assigned a larger productivity score under the hapax-based measure. The two measures agree on the high productivity of *-men* and the low productivity of *-tou*. The different results of the two measures are likely due to the type-based/token-based difference of the measures. The result of each measure requires an individual evaluation, for the knowledge that we can obtain from the result of each measure is different; for example, while the hapax-based measure takes into consideration the degree of lexicalization of words of an affix, the P_{tde} measure does not consider such an issue.

We have also examined how corpus-data variability affects the results of a productivity measure. It was found that a large number of random splits of a corpus adds stability to both the P_{tde} measure and the hapax-based measure. Moreover, it was found that even without randomization of words, the averaging of unseen types and all types under the P_{tde} measure reduces the effects of corpus-data variability. This is an advantage of the P_{tde} measure, considering the computational cost involved in randomizing words repeatedly, especially when the corpus is large.

With an assumption that unseen words of a corpus are good candidates for new words, a corpus-based productivity measurement can be regarded as a search for unseen words in a corpus. The apparent paradox is that the words that we seek are "unseen." Baayen's hapax-based measure achieves a mathematical estimate of the probability of seeing unseen words in a corpus by the Good-Turing estimation method. The deleted estimation method provides another way of defining unseen words of a corpus by comparing discrepancies in word frequency between two corpora, and the method also enables defining unseen words in a type-based context. It is hoped that words identified as unseen by the P_{tde} measure are also good candidates for new words, and this requires further investigation in future research. The implication of the successful result of the P_{tde} measure presented in this paper is that, in addition to what has been proposed by Baayen [1989, 1992, and subsequent works], there appear to be possibilities for capturing and exploiting elements in corpus data that are relevant to the quantitative description of productivity. The study of morphological productivity will be enriched by exploring such possibilities in the corpus-based approach to measuring productivity.

Acknowledgments

The author wishes to thank Harald Baayen, Richard Sproat, Martin Chodorow, and the anonymous reviewers for their insightful comments on the first draft of this paper. Any errors are the responsibility of the author.

References

Academia Sinica Balanced Corpus (Version 3.0) [CD-ROM]. Taipei, Taiwan: Academia Sinica, 1998.

Anshen, F., & Aronoff, M. "Morphological Productivity and Phonological Transparency." *Canadian Journal of Linguistics, 26,* 1981, 63–72.

Anshen, F., & Aronoff, M. "Producing Morphologically Complex Words." *Linguistics, 26,* 1988, 641–655.

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

Aronoff, M. "The Relevance of Productivity in a Synchronic Description of Word Formation." In J. Fisiak (Ed.), *Historical Morphology*. The Hague: Mouton, 1980, 71–82.

Aronoff, M. "Potential Words, Actual Words, Productivity and Frequency." *Proceedings of the International Congress of Linguists, 13,* 1983, 163–171.

Aronoff, M., & Anshen, F. "Morphology and the Lexicon: Lexicalization and Productivity." In A. Spencer & A. M. Zwicky (Eds.), *The Handbook of Morphology*. Oxford, UK: Blackwell Publishers, 1998, 237–247.

Aronoff, M., & Schvaneveldt, R. "Testing Morphological Productivity." *Annals of the New York Academy of Sciences, 318,* 1978, 106–114.

Baayen, R. H. *A Corpus-Based Study of Morphological Productivity: Statistical Analysis and Psychological Interpretation*. Doctoral dissertation, Free University, Amsterdam, 1989.

Baayen, R. H. "Quantitative Aspects of Morphological Productivity." In G. Booij & J. van Marle (Eds.), *Yearbook of Morphology 1991*. Dordrecht: Kluwer, 1992, 109–149.

Baayen, R. H. "On Frequency, Transparency and Productivity." In G. Booij & J. van Marle (Eds.), *Yearbook of Morphology 1992*. Dordrecht: Kluwer, 1993, 181–208.

Baayen, R. H. *Word Frequency Distributions*. Dordrecht: Kluwer, 2001.

Baayen, R. H., & Lieber, R. "Productivity and English Word-Formation: A Corpus-Based Study." *Linguistics, 29,* 1991, 801–843.

Baayen, R. H., & Renouf, A. "Chronicling the Times: Productive Lexical Innovations in an English Newspaper." *Language, 72,* 1996, 69–96.

Bauer, L. *Morphological Productivity*. Cambridge, UK: Cambridge University Press, 2001.

Baxter, W. H., & Sagart, L. "Word Formation in Old Chinese." In J. L. Packard (Ed.), *New Approaches to Chinese Word Formation: Morphology, Phonology and Lexicon in Modern and Ancient Chinese*. Berlin: Mouton de Gruyter, 1998, 35–76.

Booij, G. E. *Dutch Morphology: A Study of Word Formation in Generative Grammar*. Dordrecht: Foris, 1977.

Chen, P. *Modern Chinese: History and Sociolinguistics*. Cambridge University Press, 1999.

Chen, S. F., & Goodman, J. *An Empirical Study of Smoothing Techniques for Language Modeling* (Tech. Rep. No. 10-98). Cambridge, MA: Harvard University, Center for Research in Computing Technology, 1998.

Church, K. W., & Gale, W. A. "A Comparison of the Enhanced Good-Turing and Deleted Estimation Methods for Estimating Probabilities of English Bigrams." *Computer Speech and Language, 5,* 1991, 19–54.

Cutler, A. "Productivity in Word Formation." *Papers from the Sixteenth Regional Meeting of the Chicago Linguistic Society*. Chicago, IL: Chicago Linguistic Society, 1980, 45–51.

Good, I. J. "The Population Frequencies of Species and the Estimation of Population Parameters." *Biometrika, 40,* 1953, 237–264.

Guo, J. "PH: A Chinese Corpus." *Communications of COLIPS, 3* (1), 1993, 45–48.

Hockenmaier, J., & Brew, C. "Error-Driven Learning of Chinese Word Segmentation." In J. Guo, K. T. Lua, & J. Xu (Eds.), *12th Pacific Conference on Language and Information*. Singapore: Chinese and Oriental Languages Processing Society, 1998, 218–229.

Jelinek, F., & Mercer, R. "Probability Distribution Estimation for Sparse Data." *IBM Technical Disclosure Bulletin, 28,* 1985, 2591–2594.

Li, C., & Thompson, S. A. *Mandarin Chinese: A Functional Reference Grammar*. Berkeley, CA: University of California Press, 1981.

Lin, H. *A Grammar of Modern Chinese*. LINCOM EUROPA, 2001.

Manning, C. D., & Schütze, H. *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press, 1999.

Norman, J. *Chinese*. Cambridge University Press, 1988.

Packard, J. L. *The Morphology of Chinese: A Linguistic and Cognitive Approach*. Cambridge, UK: Cambridge University Press, 2000.

Plag, I. *Morphological Productivity: Structural Constraints in English Derivation*. Berlin: Mouton de Gruyter, 1999.

Ramsey, R. S. *The Languages of China*. Princeton, NJ: Princeton University Press, 1987.

Schultink, H. "Produktiviteit als Morfologisch Fenomeen." *Forum der Letteren, 2,* 1961, 110–125.

Spencer, A. *Morphological Theory: An Introduction to Word Structure in Generative Grammar*. Cambridge, UK: Cambridge University Press, 1991.

Sproat, R. "Corpus-Based Methods in Chinese Morphology." Tutorial given at COLING, Taipei, Taiwan, 2002.

Sproat, R., & Shih, C. "A Corpus-Based Analysis of Mandarin Nominal Root Compound." *Journal of East Asian Linguistics, 5,* 1996, 49–71.

Sproat, R., Shih, C., Gale, W., & Chang, N. "A Stochastic Finite-State Word-Segmentation Algorithm for Chinese." *Computational Linguistics, 22* (3), 1996, 66–73.

Uniconv [Computer Software]. Cambridge, MA: Basis Technology, 1999.

Van Marle, J. On the Paradigmatic Dimension of Morphological Productivity. Dordrecht: Foris, 1985.

Appendix: Words of the Mandarin Suffixes in the PH Corpus

Below are the words of the Mandarin suffixes and their token frequencies in the PH Corpus.

-hua

变化 *biànhuà* 495 – 现代化 *xiàndàihuà* 473 – 深化 *shēnhuà* 323 – 自由化 *zìyóuhuà* 167 – 一体化 *yītǐhuà* 138 – 强化 *qiánghuà* 131 – 恶化 *èhuà* 122 – 优化 *yōuhuà* 99 – 消化 *xiāohuà* 71 – 石化 *shíhuà* 68 – 国产化 *guóchǎnhuà* 59 – 转化 *zhuǎnhuà* 54 – 社会化 *shèhuìhuà* 53 – 正常化 *zhèngchánghuà* 52 – 美化 *měihuà* 51 – 净化 *jìnghuà* 50 – 自动化 *zìdònghuà* 50 – 电气化 *dàinqìhuà* 45 – 机械化 *jīxièhuà* 42 – 制度化 *zhìdùhuà* 41 – 标准化 *biāozhǔnhuà* 33 – 工业化 *gōngyèhuà* 29 – 氧化 *yǎnghuà* 25 – 电化 *dàinhuà* 25 – 系列化 *xìlièhuà* 22 – 民主化 *mínzhǔhuà* 22 – 科学化 *kēxuéhuà* 21 – 液化 *yèhuà* 21 – 商品化 *shāngpǐnhuà* 19 – 火化 *huǒhuà* 18 – 演化 *yǎnhuà* 18 – 革命化 *gémìnghuà* 17 – 生物化 *shēngwùhuà* 15 – 简化 *jiǎnhuà* 14 – 融化 *rónghuà* 14 – 国际化 *guójìhuà* 14 – 老化 *lǎohuà* 13 – 农机化 *nóngjīhuà* 13 – 激化 *jīhuà* 13 – 专业化 *zhuānyèhuà* 12 – 产业化 *chǎnyèhuà* 11 – 沙漠化 *shāmòhuà* 11 – 多元化 *duōyuánhuà* 10 – 裂化 *lièhuà* 10 – 军事化 *jūnshìhuà* 10 – 煤气化 *méiqìhuà* 9 – 良种化 *liángzhǒnghuà* 8 – 硬化 *yìnghuà* 8 – 生化 *shēnghuà* 8 – 法制化 *fǎzhìhuà* 8 – 分化 *fēnhuà* 8 – 林网化 *línwǎnghuà* 7 – 工厂化 *gōngchǎnghuà* 7 – 系统化 *xìtǒnghuà* 6 – 模式化 *móshìhuà* 6 – 集团化 *jítuánhuà* 6 – 大众化 *dàzhònghuà* 6 – 恐龙化 *kǒnglónghuà* 6 – 企业化 *qǐyèhuà* 6 – 殖民化 *zhímínhuà* 5 – 规模化 *guīmóhuà* 5 – 全球化 *quánqiúhuà* 5 – 活血化 *huóxuèhuà* 5 – 硫化 *liúhuà* 4 – 立体化 *lìtǐhuà* 4 – 家庭化 *jiātínghuà* 4 – 形象化 *xíngxiànghuà* 4 – 中华化 *zhōnghuáhuà* 4 – 智能化 *zhìnénghuà* 4 – 软化 *ruǎnhuà* 4 – 表面化 *biǎomiànhuà* 4 – 物化 *wùhuà* 4 – 白热化 *báirèhuà* 3 – 程序化 *chéngxùhuà* 3 – 焦化 *jiāohuà* 3 – 牙齿化 *yáchǐhuà* 3 – 纯化 *chúnhuà* 3 – 气化 *qìhuà* 3 – 园林化 *yuánlínhuà* 3 – 合作化 *hézuòhuà* 3 – 异化 *yìhuà* 3 – 风化 *fēnghuà* 3 – 焚化 *fénhuà* 3 – 资源化 *zīyuánhuà* 3 – 僵化 *jiānghuà* 3 – 作物化 *zuòwùhuà* 3 – 固化 *gùhuà* 3 – 数字化 *shùzìhuà* 3 – 歧化 *qíhuà* 2 – 遗愿化 *yíyuànhuà* 2 – 西化 *xīhuà* 2 – 集约化 *jíyuēhuà* 2 – 板化 *bǎnhuà* 2 – 化学化 *huàxuéhuà* 2 – 商业化 *shāngyèhuà* 2 – 丑化 *chǒuhuà* 2 – 反自由化 *fǎnzìyóuhuà* 2 – 区域化 *qūyùhuà* 2 – 群众化 *qúnzhònghuà* 2 – 法律化 *fǎlǜhuà* 2 – 国有化 *guóyǒuhuà* 2 – 乳化 *rǔhuà* 2 – 水利化 *shuǐlìhuà* 2 – 产品化 *chǎnpǐnhuà* 2 – 法规化 *fǎguīhuà* 2 – 基地化 *jīdìhuà* 2 – 驯化 *xúnhuà* 2 – 信息化 *xìnxīhuà* 2 – 水化 *shuǐhuà* 2 – 煤化 *méihuà* 2 – 孵化 *fūhuà* 2 – 极化 *jíhuà* 2 – 植物化 *zhíwùhuà* 2 – 中文化 *zhōngwénhuà* 2 – 资本主义化 *zīběnzhǔyìhuà* 2 – 计算机化 *jìsuànjīhuà* 2 – 电脑化 *diànnǎohuà* 1 – 短期化 *duǎnqīhuà* 1 – 赔偿仪化 *péichángyíhuà* 1 – 组织化 *zǔzhīhuà* 1 – 类型化 *lèixínghuà* 1 – 实体化 *shítǐhuà* 1 – 集体化 *jítǐhuà* 1 – 林带化 *líndàihuà* 1 – 华东化 *huádōnghuà* 1 – 湿化 *shīhuà* 1 – 鱼粉化 *yúfěnhuà* 1 – 联合化 *liánhéhuà* 1 – 批量化 *pīliànghuà* 1 – 概念化 *gàiniànhuà* 1 – 集成化 *jíchénghuà* 1 – 碱化 *jiǎnhuà* 1 – 民族化 *mínzúhuà* 1 – 管道化 *guǎndàohuà* 1 – 网络

化 *wǎngluòhuà* 1 – 氨化 *ānhuà* 1 – 整体化 *zhěngtǐhuà* 1 – 渠网化 *qúwǎnghuà* 1 – 健康化 *jiànkānghuà* 1 – 神化 *shénhuà* 1 – 本地化 *běndìhuà* 1 – 欧洲化 *ōuzhōuhuà* 1 – 合理化 *hélǐhuà* 1 – 馆化 *guǎnhuà* 1 – 规格化 *guīgéhuà* 1 – 贵族化 *guìzúhuà* 1 – 模块化 *mókuàihuà* 1 – 个性化 *gèxìnghuà* 1 – 原生动物化 *yuánshēngdòngwùhuà* 1 – 普及化 *pǔjíhuà* 1 – 成人化 *chéngrénhuà* 1 – 硬朗化 *yìnglanghuà* 1 – 欧共体化 *ōugòngtǐhuà* 1 – 氰化 *qínghuà* 1 – 定量化 *dìngliànghuà* 1 – 氯苯化 *lǜběnhuà* 1 – 电器化 *diànqìhuà* 1 – 龄化 *línghuà* 1 – 氯化 *lǜhuà* 1 – 官僚化 *guānliáohuà* 1 – 氯磺化 *lǜhuánghuà* 1 – 政治化 *zhèngzhìhuà* 1 – 关怀化 *guānhuáihuà* 1 – 档案化 *dàngànhuà* 1 – 磷化 *línhuà* 1 – 凝固化 *nínggùhuà* 1 – 质化 *zhìhuà* 1 – 溶化 *rónghuà* 1 – 皂化 *zàohuà* 1 – 尘化 *chénhuà* 1 – 藻类化 *zǎolèihuà* 1 – 元首化 *yuánshǒuhuà* 1 – 园田化 *yuántiánhuà* 1 – 腐化 *fǔhuà* 1 – 关系化 *guānxìhuà* 1 – 塑化 *sùhuà* 1 – 艺术化 *yìshùhuà* 1 – 国家化 *guójiāhuà* 1 – 足迹化 *zújìhuà* 1 – 炼化 *liànhuà* 1 – 棉花化 *miánhuahuà* 1 – 通用化 *tōngyònghuà* 1 – 渍化 *zìhuà* 1 – 行政化 *xíngzhènghuà* 1 – 越南化 *yuènánhuà* 1 – 蠕虫化 *rúchónghuà* 1 – 模硫化 *móliúhuà* 1 – 量化 *liànghuà* 1 – 时装化 *shízhuānghuà* 1 – 部门化 *bùménhuà* 1 – 理想化 *lǐxiǎnghuà* 1 – 省城化 *shěngchénghuà* 1 – 党化 *dǎnghuà* 1 – 战略化 *zhànlüèhuà* 1 – 全能化 *quánnénghuà* 1 – 催化 *cuīhuà* 1 – 数量化 *shùliànghuà* 1 – 空心化 *kòngxīnhuà* 1 – 纤化 *xiānhuà* 1 – 羽化 *yǔhuà* 1 – 套路化 *tàolùhuà* 1 – 平面化 *píngmiànhuà* 1 – 雪化 *xuěhuà* 1 – 生活化 *shēnghuóhuà* 1 – 动物化 *dòngwùhuà* 1 – 程控化 *chéngkònghuà* 1 – 氮化 *dànhuà* 1 – 谱化 *pǔhuà* 1 – 庸俗化 *yōngsúhuà* 1

-men

人们 *rénmen* 734 – 代表们 *dàibiǎomen* 175 – 专家们 *zhuānjiāmen* 117 – 委员们 *wěiyuánmen* 109 – 工人们 *gōngrénmen* 75 – 同志们 *tóngzhìmen* 72 – 孩子们 *háizimen* 64 – 战士们 *zhànshìmen* 59 – 职工们 *zhígōngmen* 39 – 同学们 *tóngxuémen* 32 – 队员们 *duìyuánmen* 31 – 姑娘们 *gūniangmen* 26 – 客人们 *kèrenmen* 24 – 记者们 *jìzhěmen* 23 – 科学家们 *kēxuéjiāmen* 23 – 老人们 *lǎorénmen* 23 – 农民们 *nóngmínmen* 22 – 学生们 *xuéshengmen* 21 – 分析家们 *fēnxījiāmen* 21 – 姐妹们 *jiěmèimen* 19 – 朋友们 *péngyoumen* 18 – 艺术家们 *yìshùjiāmen* 16 – 干部们 *gànbùmen* 16 – 市民们 *shìmínmen* 15 – 市长们 *shìzhǎngmen* 14 – 居民们 *jūmínmen* 14 – 首脑们 *shǒunǎomen* 14 – 村民们 *cūnmínmen* 13 – 演员们 *yǎnyuánmen* 13 – 旅客们 *lǚkèmen* 12 – 同事们 *tóngshìmen* 12 – 小伙子们 *xiǎohuǒzimen* 11 – 医生们 *yīshēngmen* 10 – 行家们 *xíngjiāmen* 10 – 议员们 *yìyuánmen* 10 – 大学生们 *dàxuéshēngmen* 10 – 官兵们 *guānbīngmen* 9 – 运动员们 *yùndòngyuánmen* 9 – 观察家们 *guānchájiāmen* 9 – 同行们 *tóngxíngmen* 8 – 经理们 *jīnglǐmen* 8 – 师生们 *shīshēngmen* 7 – 常委们 *chángwěimen* 7 – 企业家们 *qǐyèjiāmen* 7 – 外长们 *wàizhǎngmen* 7 – 指战员们 *zhǐzhànyuánmen* 7 – 船员们 *chuányuánmen* 6 – 列车员们 *lièchēyuánmen* 6 – 部长们 *bùzhǎngmen* 6 – 作家们 *zuòjiāmen* 6 – 建设者们 *jiànshèzhěmen* 6 – 工友们 *gōngyǒumen* 6 – 青年们 *qīngniánmen* 6 – 党员们 *dǎngyuánmen* 5 – 顾客们 *gùkèmen* 5 – 干警们 *gànjǐngmen* 5 – 学者们 *xuézhěmen* 5 – 娘们 *niángmen* 5 – 劳模们 *láomómen* 5 – 教师们 *jiàoshīmen* 5 – 营业员们 *yíngyèyuánmen*

4 – 团员们 *tuányuánmen* 4 – 成员们 *chéngyuánmen* 4 – 子女们 *zǐnǚmen* 4 – 队友们 *duìyǒumen* 4 – 妇女们 *fùnǚmen* 4 – 乘客们 *chéngkèmen* 4 – 侨胞们 *qiáobāomen* 4 – 伙伴们 *huǒbànmen* 4 – 来宾们 *láibīnmen* 4 – 儿女们 *érnǚmen* 3 – 军人们 *jūnrénmen* 3 – 将军们 *jiāngjūnmen* 3 – 父母官们 *fùmǔguānmen* 3 – 乘务员们 *chéngwùyuánmen* 3 – 护士们 *hùshimen* 3 – 大师们 *dàshīmen* 3 – 儿孙们 *érsūnmen* 3 – 戏迷们 *xìmímen* 3 – 小学生们 *xiǎoxuéshēngmen* 3 – 文艺家们 *wényìjiāmen* 3 – 观众们 *guānzhòngmen* 3 – 球迷们 *qiúmímen* 3 – 司长们 *sīchángmen* 3 – 领导们 *lǐngdǎomen* 3 – 教练员们 *jiàoliànyuánmen* 2 – 爷们 *yémen* 2 – 人员们 *rényuánmen* 2 – 女工们 *nǚgōngmen* 2 – 摄影家们 *shèyǐngjiāmen* 2 – 板报员们 *bǎnbàoyuánmen* 2 – 老板们 *lǎobǎnmen* 2 – 老汉们 *lǎohànmen* 2 – 状元们 *zhuàngyuanmen* 2 – 会员们 *huìyuánmen* 2 – 州长们 *zhōuzhǎngmen* 2 – 女士们 *nǚshìmen* 2 – 友人们 *yǒurénmen* 2 – 大家们 *dàjiāmen* 2 – 师傅们 *shīfumen* 2 – 创作者们 *chuàngzuōzhěmen* 2 – 喇嘛们 *lǎmamen* 2 – 经济学家们 *jīngjìxuéjiāmen* 2 – 支持者们 *zhīchízhěmen* 2 – 老师们 *lǎoshīmen* 2 – 儿子们 *érzimen* 2 – 祖辈们 *zǔbèimen* 2 – 少女们 *shàonǚmen* 2 – 学员们 *xuéyuánmen* 2 – 书画家们 *shūhuàjiāmen* 2 – 选手们 *xuǎnshǒumen* 2 – 妈妈们 *māmamen* 2 – 同胞们 *tóngbāomen* 2 – 员工们 *yuángōngmen* 2 – 亲戚们 *qīnqimen* 2 – 选民们 *xuǎnmínmen* 2 – 天文学家们 *tiānwénxuéjiāmen* 2 – 儿童们 *értóngmen* 2 – 法官们 *fǎguānmen* 1 – 行人们 *xíngrénmen* 1 – 歹徒们 *dǎitúmen* 1 – 高徒们 *gāotúmen* 1 – 瘾君子们 *yǐnjūnzǐmen* 1 – 贵宾们 *guìbīnmen* 1 – 厨师们 *chúshīmen* 1 – 台胞们 *táibāomen* 1 – 老伙伴们 *lǎohuǒbànmen* 1 – 勇士们 *yǒngshìmen* 1 – 车迷们 *chēmímen* 1 – 支委们 *zhīwěimen* 1 – 孙子们 *sūnzimen* 1 – 夫妇们 *fūfùmen* 1 – 配水员们 *pèishuǐyuánmen* 1 – 伤员们 *shāngyuánmen* 1 – 囚犯们 *qiúfànmen* 1 – 客户们 *kèhùmen* 1 – 军官们 *jūnguānmen* 1 – 士兵们 *shìbīngmen* 1 – 巾帼们 *jīnguómen* 1 – 助手们 *zhùshǒumen* 1 – 留学生们 *liúxuéshēngmen* 1 – 设计师们 *shèjìshīmen* 1 – 局长们 *júzhǎngmen* 1 – 老工人们 *lǎogōngrénmen* 1 – 渔工们 *yúgōngmen* 1 – 副市长们 *fùshìzhǎngmen* 1 – 侦察员们 *zhēncháyuánmen* 1 – 观察员们 *guāncháyuánmen* 1 – 设计者们 *shèjìzhěmen* 1 – 家属们 *jiāshǔmen* 1 – 检察官们 *jiǎncháguānmen* 1 – 体育迷们 *tǐyùmímen* 1 – 女生们 *nǚshēngmen* 1 – 革命先烈们 *gémìngxiānlièmen* 1 – 飞行员们 *fēixíngyuánmen* 1 – 老头子们 *lǎotóuzimen* 1 – 海外侨胞们 *hǎiwàiqiáobāomen* 1 – 炮制者们 *pàozhìzhěmen* 1 – 服务员们 *fúwùyuánmen* 1 – 推销员们 *tuīxiāoyuánmen* 1 – 太太们 *tàitaimen* 1 – 伐木者们 *fámùzhěmen* 1 – 劳动模范们 *láodòngmófànmen* 1 – 水兵们 *shuǐbīngmen* 1 – 使节们 *shǐjiémen* 1 – 歌唱家们 *gēchàngjiāmen* 1 – 主任们 *zhǔrènmen* 1 – 个体户们 *gètǐhùmen* 1 – 演说家们 *yǎnshuōjiāmen* 1 – 音乐家们 *yīnyuèjiāmen* 1 – 亲友们 *qīnyǒumen* 1 – 功臣们 *gōngchénmen* 1 – 职员们 *zhíyuánmen* 1 – 姐姐们 *jiějiemen* 1 – 司机们 *sījīmen* 1 – 制造商们 *zhìzàoshāngmen* 1 – 英雄们 *yīngxióngmen* 1 – 画家们 *huàjiāmen* 1 – 外商们 *wàishāngmen* 1 – 患者们 *huànzhěmen* 1 – 村邻们 *cūnlínmen* 1 – 卫士们 *wèishìmen* 1 – 大臣们 *dàchénmen* 1 – 技术员们 *jìshùyuánmen* 1 – 图者们 *túzhěmen* 1 – 教员们 *jiàoyuánmen* 1 – 老大娘们 *lǎodàniángmen* 1 – 法学家们 *fǎxuéjiāmen* 1 – 研究者们

yánjiūzhěmen 1 – 游人们 *yóurénmen* 1 – 元首们 *yuánshǒumen* 1 – 娃娃们 *wáwamen* 1 – 青少年们 *qīngshàoniánmen* 1 – 力士们 *lìshìmen* 1 – 售货员们 *shòuhuòyuánmen* 1 – 教练们 *jiàoliànmen* 1 – 采购员们 *cǎigòuyuánmen* 1 – 女们 *nǚmen* 1 – 游客们 *yóukèmen* 1 – 烈士们 *lièshìmen* 1 – 西藏史学家们 *xīzàngshǐxuéjiāmen* 1 – 老奶奶们 *lǎonǎinaimen* 1 – 大夫们 *dàifūmen* 1 – 气象学家们 *qìxiàngxuéjiāmen* 1 – 工作者们 *gōngzuòzhěmen* 1 – 县太爷们 *xiàntàiyémen* 1 – 商贩们 *shāngfànmen* 1 – 松们 *sōngmen* 1 – 亲人们 *qīnrénmen* 1 – 老朋友们 *lǎopéngyoumen* 1 – 家长们 *jiāzhǎngmen* 1 – 夫妻们 *fūqīmen* 1 – 学子们 *xuézǐmen* 1 – 东道主们 *dōngdàozhǔmen* 1 – 省长们 *shěngzhǎngmen* 1 – 同仁们 *tóngrénmen* 1 – 山水画家们 *shānshuǐhuàjiāmen* 1 – 战略家们 *zhànlüèjiāmen* 1 – 董事长们 *dǒngshìzhǎngmen* 1

-r

这儿 *zhèr* 32 – 会儿 *huìr* 30 – 哪儿 *nǎr* 18 – 劲儿 *jìnr* 13 – 事儿 *shìr* 12 – 点儿 *diǎnr* 9 – 那儿 *nàr* 8 – 伙儿 *huǒr* 7 – 个儿 *gèr* 7 – 活儿 *huór* 5 – 鸟儿 *niǎor* 5 – 块儿 *kuàir* 4 – 花儿 *huār* 3 – 法儿 *fǎr* 3 – 风儿 *fēngr* 2 – 字儿 *zìr* 2 – 条儿 *tiáor* 2 – 味儿 *wèir* 2 – 片儿 *piànr* 2 – 玩儿 *wánr* 2 – 弯儿 *wānr* 2 – 样儿 *yàngr* 1 – 轧伙儿 *yàhuǒr* 1 – 脸儿 *liǎnr* 1 – 干劲儿 *gānjìnr* 1 – 头儿 *tóur* 1 – 万儿 *wànr* 1 – 话儿 *huàr* 1 – 抠儿 *kōur* 1 – 犟劲儿 *jiàngjìnr* 1 – 信儿 *xìnr* 1 – 塞儿 *sèr* 1 – 主儿 *zhǔr* 1 – 芯儿 *xīnr* 1 – 当儿 *dāngr* 1

-tou

势头 *shìtou* 133 – 码头 *mǎtou* 99 – 街头 *jiētóu* 96 – 石头 *shítou* 33 – 罐头 *guàntou* 30 – 镜头 *jìngtóu* 26 – 年头 *niántóu* 20 – 拳头 *quántou* 18 – 馒头 *mántou* 16 – 炕头 *kàngtóu* 14 – 老头 *lǎotóu* 12 – 心头 *xīntóu* 11 – 木头 *mùtou* 9 – 骨头 *gǔtou* 9 – 源头 *yuántóu* 8 – 口头 *kǒutóu* 8 – 苗头 *miáotou* 7 – 地头 *dìtóu* 7 – 指头 *zhǐtou* 7 – 锄头 *chútou* 5 – 桥头 *qiáotóu* 5 – 部头 *bùtóu* 4 – 枕头 *zhěntou* 3 – 斧头 *fǔtou* 2 – 先头 *xiāntóu* 2 – 脚趾头 *jiǎozhǐtou* 2 – 里头 *lǐtou* 2 – 风头 *fēngtou* 2 – 手指头 *shǒuzhǐtóu* 2 – 犁头 *lítóu* 2 – 滩头 *tāntóu* 1 – 丫头 *yātou* 1 – 窝窝头 *wōwōtóu* 1 – 关头 *guāntóu* 1 – 眉头 *méitóu* 1 – 两头 *liǎngtóu* 1

-zi

孩子 *háizi* 457 – 种子 *zhǒngzi* 146 – 儿子 *érzi* 131 – 日子 *rìzi* 129 – 妻子 *qīzi* 112 – 班子 *bānzi* 105 – 路子 *lùzi* 63 – 篮子 *lánzi* 58 – 伙子 *huǒzi* 53 – 房子 *fángzi* 50 – 帽子 *màozi* 37 – 一下子 *yíxiàzi* 29 – 样子 *yàngzi* 27 – 辈子 *bèizi* 25 – 饺子 *jiǎozi* 23 – 贩子 *fànzi* 22 – 担子 *dànzi* 21 – 孙子 *sūnzi* 20 – 牌子 *páizi* 20 – 肚子 *dùzi* 19 – 步子 *bùzi* 18 – 村子 *cūnzi* 18 – 一揽子 *yīlǎnzǐ* 16 – 桔子 *júzi* 16 – 脖子 *bózi* 15 – 身子 *shēnzǐ* 14 – 竹子 *zhúzi* 12 – 汉子 *hànzi* 11 – 侄子 *zhízi* 10 – 车子 *chēzi* 10 – 钉子 *dīngzi* 10 – 屋子 *wūzi* 10 – 厂子 *chǎngzi* 10 – 册子 *cèzi* 9 – 鼻子 *bízi* 9 – 茄子 *qiézi* 9 – 粒子 *lìzi* 8 – 苗子 *miáozi* 8 – 裙子 *qúnzi* 8 – 脑子 *nǎozi* 8 – 林子 *línzi* 8 – 椅子 *yǐzi* 8 – 鸽子 *gēzi* 8 – 被子 *bèizi* 8 – 鞋子 *xiézi* 7 – 沙子 *shāzi* 7 – 西门子 *xīménzi* 7 – 幌子 *huǎngzi*

6 – 绳子 *shéngzi* 6 – 袋子 *dàizi* 6 – 金子 *jīnzi* 6 – 影子 *yǐngzi* 6 – 例子 *lìzi* 6 – 枪杆子 *qiānggānzi* 6 – 斧子 *fǔzi* 6 – 口子 *kǒuzi* 6 – 梆子 *bāngzi* 5 – 底子 *dǐzi* 5 – 袜子 *wàzi* 5 – 膀子 *bǎngzi* 5 – 嗓子 *sǎngzi* 5 – 桌子 *zhuōzi* 5 – 票子 *piàozi* 5 – 胡子 *húzi* 5 – 话匣子 *huàxiázi* 5 – 圈子 *quānzi* 4 – 摊子 *tānzi* 4 – 棍子 *gùnzi* 4 – 杆子 *gānzi* 4 – 园子 *yuánzi* 4 – 院子 *yuànzi* 4 – 炉子 *lúzi* 4 – 果子 guǒzi 4 – 筷子 *kuàizi* 4 – 豹子 *bàozi* 4 – 片子 *piànzi* 4 – 刀子 *dāozi* 4 – 箱子 *xiāngzi* 3 – 匣子 *xiázi* 3 – 裤子 *kùzi* 3 – 褥子 *rùzi* 3 – 瓶子 *píngzi* 3 – 胆子 *dǎnzi* 3 – 豆子 *dòuzi* 3 – 个子 *gèzi* 3 – 点子 diǎnzi 3 – 狮子 shīzi 3 – 阵子 zhènzi 3 – 小子 xiǎozi 3 – 老头子 *lǎotóuzi* 3 – 台子 *táizi* 3 – 叶子 *yèzi* 3 – 杯子 *bēizi* 3 – 帘子 *liánzi* 2 – 梯子 *tīzi* 2 – 烂摊子 *làntānzi* 2 – 毯子 *tǎnzi* 2 – 瞎子 *xiāzi* 2 – 毽子 *jiànzi* 2 – 燕子 *yànzi* 2 – 兔子 *tùzi* 2 – 袖子 *xiùzi* 2 – 椰子 *yēzi* 2 – 瘤子 *liúzi* 2 – 猴子 *hóuzi* 2 – 盒子 *hézi* 2 – 虫子 *chóngzi* 2 – 蝎子 *xiēzi* 2 – 案子 *ànzi* 2 – 句子 *jùzi* 2 – 模子 *mózi* 2 – 空子 *kòngzi* 2 – 鞭子 *biānzi* 2 – 命根子 *mìnggēnzi* 2 – 曲子 *qǔzi* 2 – 法子 *fǎzi* 1 – 窗子 *chuāngzi* 1 – 谷子 *gǔzi* 1 – 哨子 *shàozi* 1 – 靶子 *bǎzi* 1 – 凳子 *jīzi* 1 – 兜子 *dōuzi* 1 – 尖子 *jiānzi* 1 – 岔子 *chàzi* 1 – 游子 *yóuzi* 1 – 老样子 *lǎoyàngzi* 1 – 褂子 *guàzi* 1 – 乱子 *luànzi* 1 – 苇子 *wěizi* 1 – 坝子 *bàzi* 1 – 空架子 *kōngjiàzi* 1 – 银子 *yínzi* 1 – 阀子 *fázi* 1 – 丸子 *wánzi* 1 – 笛子 *dízi* 1 – 棚子 *péngzi* 1 – 辫子 *biànzi* 1 – 栗子 *lìzi* 1 – 柿子 *shìzi* 1 – 链子 *liànzi* 1 – 头子 *tóuzi* 1 – 蹄子 *tízi* 1 – 梭子 *suōzi* 1 – 骡子 *luózi* 1 – 骗子 *piànzi* 1 – 柚子 *yòuzi* 1 – 锤子 *chuízi* 1 – 石磙子 *shígǔnzi* 1 – 箕子 *jīzi* 1 – 槽子 *cáozi* 1 – 锭子 *dìngzi* 1 – 两口子 *liǎngkǒuzi* 1 – 椽子 *chuánzi* 1 – 单子 *dānzi* 1 – 剪子 *jiǎnzi* 1 – 档子 *dàngzi* 1 – 沙苑子 *shāyuànzǐ* 1 – 面子 *miànzi* 1 – 缨子 *yīngzi* 1 – 号子 *hàozi* 1 – 皮夹子 *píjiāzi* 1 – 镯子 *zhuózi* 1 – 卒子 *zúzi* 1 – 橙子 *chéngzi* 1 – 集子 *jízi* 1 – 鼓子 *gǔzi* 1 – 扇子 *shānzi* 1 – 桶子 *tǒngzi* 1 – 桃子 *táozi* 1 – 脚脖子 *jiǎobózi* 1 – 叔子 *shūzi* 1 – 庄子 *zhuāngzi* 1 – 胖子 *pàngzi* 1 – 杏子 *xìngzi* 1 – 狍子 *páozi* 1 – 台柱子 *táizhùzi* 1 – 份子 *fènzi* 1

Computational Linguistics and Chinese Language Processing
Vol. 8, No. 1, February 2003 , pp. 77-102

Extension of Zipf's Law to Word and Character *N*-grams for English and Chinese

Le Quan Ha*, E. I. Sicilia-Garcia*, Ji Ming* and F. J. Smith*

Abstract

It is shown that for a large corpus, Zipf 's law for both words in English and characters in Chinese does not hold for all ranks. The frequency falls below the frequency predicted by Zipf's law for English words for rank greater than about 5,000 and for Chinese characters for rank greater than about 1,000. However, when single words or characters are combined together with *n*-gram words or characters in one list and put in order of frequency, the frequency of tokens in the combined list follows Zipf's law approximately with the slope close to -1 on a log-log plot for all *n*-grams, down to the lowest frequencies in both languages. This behaviour is also found for English 2-byte and 3-byte word fragments. It only happens when all *n*-grams are used, including semantically incomplete *n*-grams. Previous theories do not predict this behaviour, possibly because conditional probabilities of tokens have not been properly represented.

Keywords: Zipf 's law, Chinese character, Chinese compound word, *n*-grams, phrases.

1. Introduction

1.1 Zipf's law

The law discovered empirically by [Zipf 1949] for word tokens in a corpus states that if *f* is the frequency of a word in the corpus and *r* is the rank, then:

$$f = \frac{k}{r} \qquad (1)$$

where *k* is a constant for the corpus. When *log(f)* is drawn against *log(r)* in a graph (which is

* Computer Science School, Queen's University Belfast, Belfast BT7 1NN, Northern Ireland, UK.
Email: {q.le, e.sicilia, j.ming, fj.smith}@qub.ac.uk

called a Zipf curve), a straight line is obtained with a slope of –1. An example with a small corpus of 250,000 tokens made up of paragraphs chosen at random from the Brown corpus of American English [Francis and Kucera 1964] is given in Figure 1; in this the tokens do not include punctuation marks and numbers. Typographical errors, if any, will appear in the hapax legomenon.

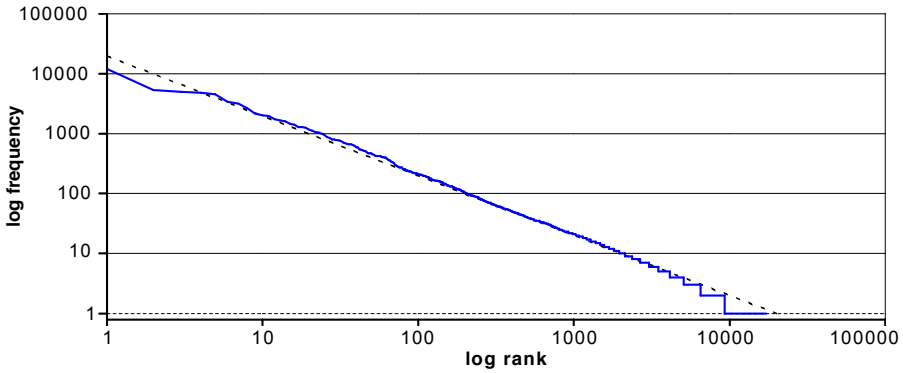

Figure 1 Zipf curve for the unigrams extracted from a 250,000-word tokens corpus.

Zipf's discovery was followed by a large body of literature, reviewed in a series of papers edited by [Guiter and Arapov 1982]. Notable among these are papers by [Mandelbrot 1953, 1954, 1959, 1961], [Miller 1954, 1957, 1958], [Simon 1955, 1960, 1961], [Sichel 1975, 1986], [Carroll 1967, 1969], [Baayen 1991], [Chitashvili 1983, 1989] and [Orlov 1983]. It continues to stimulate interest today [Samuelson 1996]; [Baayen 2001]; [Hatzigeorgiu, Mikros and Carayannis 2001]; [Montermurro 2001]; [Ferrer and Solé 2002] and, for example, it has been recently applied to citations [Silagadze 1997], to biological species-abundance [Sichel 1997] and to DNA sequences [Yonezawa and Motohasi 1999]; [Li 2001].

Zipf discovered the law by analysing manually the frequencies of words in the novel "Ulysses" by James Joyce. It contains a vocabulary of 29,899 different word types associated with 260,430 word tokens.

1.2 Theoretical developments:

Following its discovery in 1949, several experiments aided by the appearance of the computer in the 1960's, confirmed that the law was correct for the small corpora that could be processed at that time. The slope of the curve was found to vary slightly from –1 for some corpora; also the frequencies for the highest ranked words sometimes deviated from the straight line, which

suggested several modifications of the law, and in particular one derived theoretically by [Mandelbrot 1953] with the form:

$$f = \frac{k}{(r+\alpha)^{\beta}} \qquad (2)$$

where α and β are constants for the corpus being analysed. However, generally the constants α and β were found to be only small varying deviations from the original law by Zipf. Exceptions include legal texts which have smaller slopes (≈0.9) showing that lawyers use more word types than other people! [Smith and Devine 1985].

A number of theoretical explanations for Zipf's law had been derived, many reviewed by [Fedorowicz 1982]; notably are those due to [Mandelbrot 1954, 1957], [Miller 1954, 1958], [Simon 1955], [Booth 1967], and [Sichel 1975, 1986]. Simon's derivation was controversial and a correspondence in the scientific press developed between Mandelbrot and Simon on the validity of this derivation (1959-1961); the dispute was not resolved by the time Zipf curves for larger corpora were beginning to be computed.

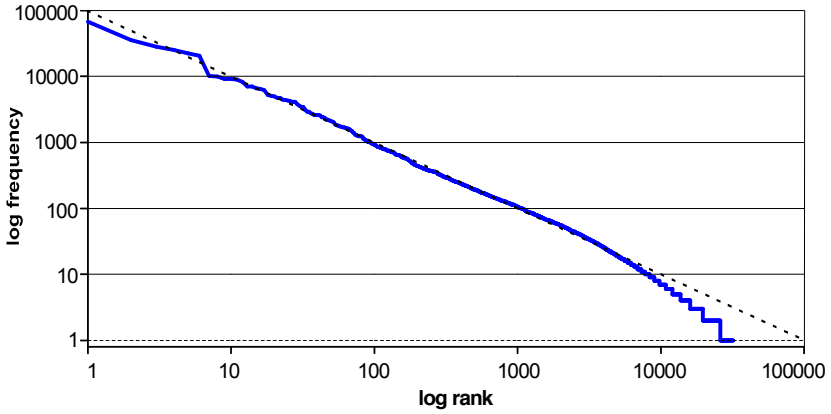

Figure 2 Zipf curve for the unigrams extracted from the 1 million words of the Brown corpus showing that the Zipf curve falls below the line with slope -1 for rank > 5,000.

The processing of larger corpora with 1 million words or more was facilitated by the development of PC's in the 1980's. When Zipf curves for these corpora were drawn, they were found to drop below the Zipf straight line with slope of –1 at the bottom of the curve, for rank greater than about 5,000. This is illustrated in Figure 2, which shows the Zipf curve for the whole of the Brown corpus (1 million words), again excluding punctuations and numbers.

This deviation from Zipf 's law was interpreted for single-author texts to represent the limited numbers of words in each author's diction. But we see in Figure 2 that a deviation also occurs for a multi-author corpus covering a wide range of domains such as the Brown corpus; so the drop in the curve is not likely to be only due to the limited number of words.

2. Zipf curves for large English corpora

We are going to explore the above deviation from Zipf's law for large corpora in two languages: Chinese and English. We begin with English.

2.1 Single words

The English corpora used in our experiments are the full text of articles appearing in the Wall Street Journal [Paul and Baker 1992] for 1987, 1988, 1989, with sizes approximately 19 million, 16 million and 6 million tokens respectively. The Zipf curves for the 3 corpora are shown in Figure 3.

For pre-execution of this corpus, numbers were written as words, e.g. 23 became "twenty three" and punctuation marks were excluded. The characters "=", "#", "~", "<", ">", "/", "+", "-", "^", "*", "@", "/" and "\", etc. were also ignored.

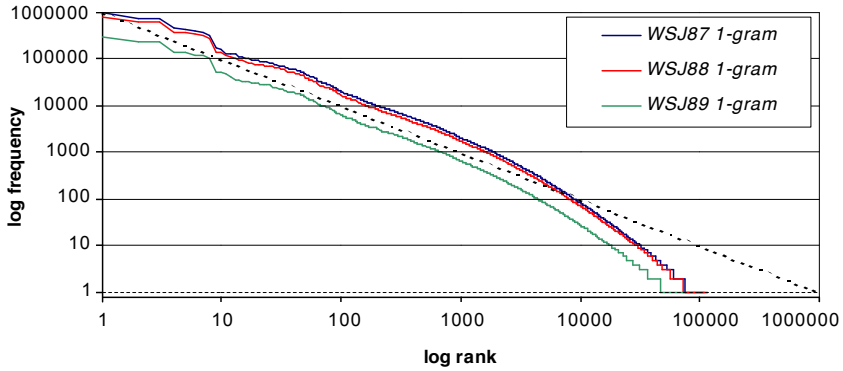

Figure 3 Zipf curves for the unigrams extracted from the 3 training corpora of WSJ

The Zipf curves for the three corpora are parallel, showing similar structures with all 3 curves deviating from Zipf's law for larger r in exactly the same way as the curve for the Brown corpus. Their separation is due to their different sizes.

2.2 *n*-Grams

Language is not made of individual words, each with its own separate piece of information, but consists of sequences of words, made up of individual words and of phrases of 2, 3 or more words together called *n*-grams. So it is interesting to measure the frequencies of *n*-grams and draw the corresponding Zipf curves.

To do this we allowed *n*-grams to overlap. For example, for the sentence: "*The cat sat on the mat*", there are four trigrams: (1) "*the cat sat*", (2) "*cat sat on*", (3) "*sat on the*" and (4) "*on the mat*". So semantically incomplete *n*-grams such as "*cat sat on*" are included in our study. No *n*-gram crossed over a punctuation mark. So a fullstop, comma, colon, etc. always ends an *n*-gram and a new *n*-gram starts after the punctuation. Thus the sentence "*Three blind mice, see how they run*" has only three trigrams "*three blind mice*", "*see how they*" and "*how they run*".

For each value of *n* between 2 and 5, we thus computed the frequencies of all *n*-grams in each corpus and put them in rank order as we had done for the words. This enabled us to draw the Zipf curves for 2-, 3-, 4- and 5-grams which are shown along with the single word curves in Figure 4, Figure 5 and Figure 6 for the three corpora. These curves are similar to the first Zipf curves drawn for *n*-grams by [Smith and Devine 1985]; but these earlier curves were for a much smaller corpus.

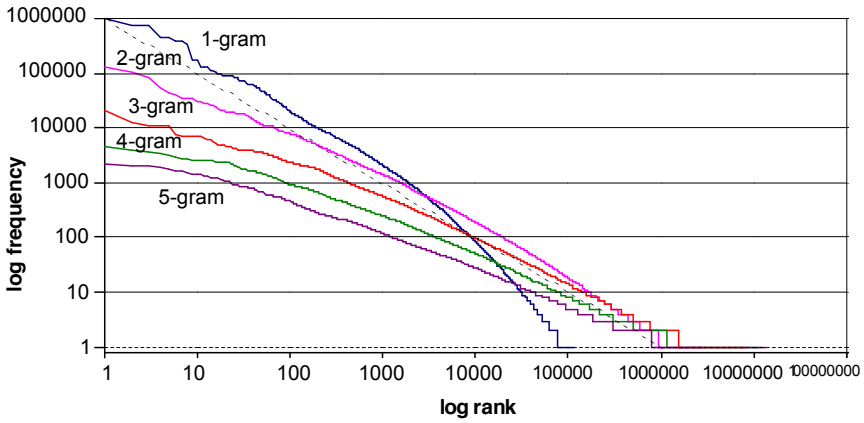

Figure 4 Zipf curves for the WSJ87 corpus

The *n*-gram Zipf curves do not follow straight lines but curve gently downwards. The average slope decreases from about 0.66 for the bigrams to about 0.59 for the 5-grams.

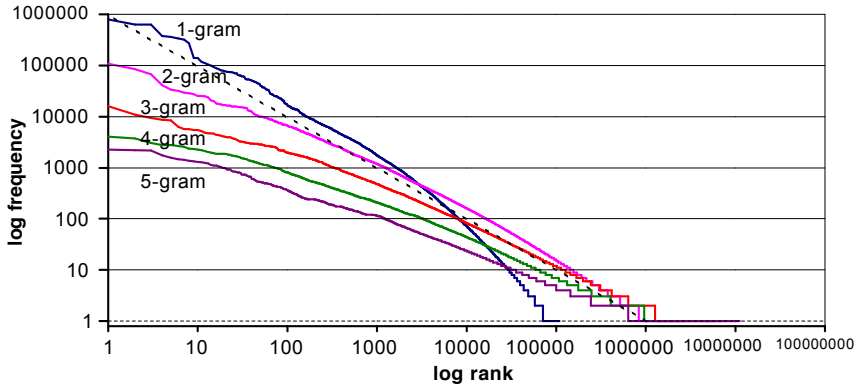

Figure 5 Zipf curves for the WSJ88 corpus

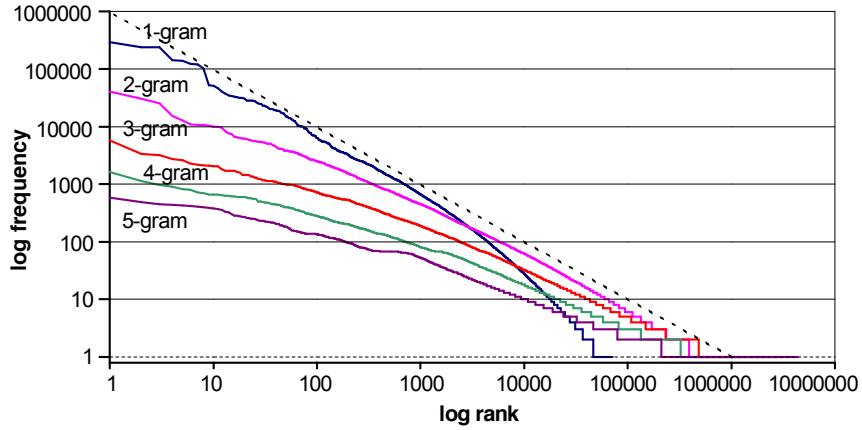

Figure 6 Zipf curves for the WSJ89 corpus

First for WSJ87, the crossing point between the unigram and bigram curves is at rank 2,943 and for the unigram and trigram curves, it is at rank 8,497. For WSJ88, these crossing points are similar, at rank 2,913 and at rank 8,404, and for WSJ89, they are at rank 2,908 and 7,960. So the unigram curves cross the bigram curves when the rank ≅ 3,000 in all 3 cases, and for the unigram and trigram curves, they cross at rank ≅ 8,000.

The ten most common words, bigrams and trigrams in the combined WSJ corpus of 40 million words are listed in Table 1.

Table 1. The most common unigrams, bigrams and trigrams in the combined WSJ

Unigrams		Bigrams		Trigrams	
Frequency	Token	Frequency	Token	Frequency	Token
2,057,968	the	217,427	of the	42,030	the U. S.
973,650	of	173,797	in the	27,260	in nineteen eighty
940,525	to	110,291	million dollars	24,165	cents a share
853,342	a	89,184	U. S.	18,233	nineteen eighty six
825,489	and	83,799	nineteen eighty	16,786	nineteen eighty seven
711,462	in	76,187	for the	15,316	five million dollars
368,012	that	72,312	to the	14,943	million dollars or
362,771	for	65,565	on the	14,517	million dollars in
298,646	one	63,838	one hundred	12,327	in New York
281,190	is	55,014	that the	11,981	a year earlier

2.3 Hapax legomena and dis legomena

The size of the hapax legomena (tokens with frequency 1) for the *n*-grams rises rapidly with *n* as shown in Table 2a, but it can not rise above the number of tokens; so the rate of increase has slowed when $n = 5$ since almost all tokens are in the hapax legomena. The hapax dis legomena (tokens with frequency 2) is much smaller and reaches a maximum for trigrams from all 3 corpora (see Table 2b) because almost all of the tokens have frequency 1, leaving a smaller number with frequency 2 when $n = 4$ and 5.

Table 2a) Number of hapax legomena for the English corpora.

Corpus		WSJ87	WSJ88	WSJ89
No of Tokens		18,790,794	15,757,051	5,946,585
No of Types		114,581	108,522	71,837
Hapax legomena	Unigram	38,853	36,945	25,162
	Bigram	1,786,290	1,620,385	851,542
	Trigram	6,601,243	5,799,257	2,598,509
	4-gram	10,635,310	9,137,402	3,736,880
	5-gram	12,493,656	10,612,036	4,376,741

Table 2b) Number of hapax dis legomena for the English corpora.

Corpus		WSJ87	WSJ88	WSJ89
No of Tokens		18,790,794	15,757,051	5,946,585
No of Types		114,581	108,522	71,837
Hapax dis- -legomena	Unigram	14,855	14,431	9,861
	Bigram	349,205	314,496	155,068
	Trigram	742,771	632,372	251,435
	4-gram	670,106	546,951	190,947
	5-gram	485,487	389,113	130,544

2.4 The nature of *n*-grams

It can be argued that most of the *n*-grams in the hapax legomena or hapax dis legomena are not meaningful, since they are semantically incomplete. Certainly that meaning may be incomplete and they need the words on either side of them to realise their full meaning. But then it can be argued that this is true of every *n*-gram (and indeed for every word). So we take the view that every *n*-gram taken from a natural language text produced by humans has meaning, though often incomplete.

However, Miller's monkey typing on a word typewriter would produce mainly meaningless *n*-grams, e.g. "*the the the*", as well as those others which have meaning by accident. The number of possible *n*-grams which the monkey can type is huge. For example, for the WSJ87 corpus there are more than 10^{15} possible trigrams of which less than 7 million produced by humans appear in the Hapax legomenon for the corpus.

Whatever one's views on the meaning of some of these incomplete *n*-grams, we report in this paper on the Zipf curves for all *n*-grams in a corpus. A later paper will include discussion on the equivalent curves for semantically complete phrases.

One of our reasons for including all *n*-grams is that statistical language modellers have been using *n*-grams, similar to the ones we have defined, which include semantically incomplete *n*-grams, with great success in modelling language over the last 20 years [Jelinek and Mercer 1985]; [O'Boyle, Owens and Smith 1994]; [Ney 1999].

3. Zipf Curves for Chinese Corpora

In Chinese, compound words can be created, made up of two or more characters. However, it is not always easy to automatically segment a written sentence in Chinese into compound

words as these are not separated by spaces as in English. Nevertheless, the extraction of a word sequence from a Chinese document has been the subject of study by many authors [Zhu 1981]; [Chen and Shi 1992]; [Bates, Chen, Li, Opie and Tzeng 1993]; [Packard 2000]; [Sproat 2002]; [Tsai and Hsu 2002]; who reference other papers.

Unfortunately, there is still ambiguity in the process of compound word extraction. For example, the following string of characters can be broken into the words: 北京 (Beijing) 城 (city) 里 (in) 交通 (traffic) 繁忙 (busy) (*The traffic in Beijing is very busy*) or into the words 北 (North) 京城 (capital city) 里 (in) 交通 (traffic) 繁忙 (busy) (*The traffic in the north of the capital city is very busy*). Only a human can distinguish which is correct, another example is: 上海 (Shanghai) 边 建设 (build) 边 发展 (develop) (*Shanghai is developing while it is building (up)*) which can also be interpreted as 上 (go to) 海边 (seaside) 建设 边 发展 (*Go to seaside to develop and build*). Once again a human is needed to decide the meaning.

Therefore, it is difficult to write a computer program to extract the correct word sequence, and for a corpus of 250 million syllables, it is impossible to do by hand. So we proceeded as follows: first of all, we used a 50,000 word-syllable dictionary (which can be found at http://www.euroasiasoftware.com/), but the extraction of the words from the text is still partly ambiguous. When a sequence of syllables was found that matched a word in the dictionary, it was usually accepted as a word. When an ambiguity occurs, e.g. 暴风骤雨 which can be one word *hurricane*, or two bi-syllable words: 暴风 骤雨 *storm shower*, then the longer word was accepted 暴风骤雨 *hurricane*. Similarly, 百万富翁 *millionaire* is accepted as one word instead of the three words 百万 富 翁 *million(s) rich elder*.

Although the whole corpus could not be checked manually, the higher frequency *n*-grams can be checked, for example the following 6-gram has been broken into the pattern: 埃及 总统 穆 巴 拉 克 rather than the pattern 埃及 总统 穆巴拉克 *Egyptian president Mubarak*. This occurs 1,865 times and could be corrected for all 1,865 occurrences in one step all over the corpus. Another example is the 7-gram: 阿 联 酋 乌 兹 别 克 occurring 7 times which should be the 2 names 阿联酋 乌兹别克 *Alanqiu Wuzibieke* to be correct. Because of the multiple occurrence of *n*-grams all of which can be corrected by one change, this speeded-up the manual process considerably. Checking all of the high frequency *n*-grams took more than 2 months work; after this, a check on a test text of 3,117 tokens was found to have 82 errors (2.6%) by an independent native speaker (other than the authors), which we took as acceptable. (The corpus can be made available on request to q.le@qub.ac.uk or fj.smith@qub.ac.uk).

Two corpora were used in our experiments: the TREC corpus and the Mandarin Daily

News corpus. Both are from the Linguistic Data Consortium[1].

There is a small overlap between the Chinese TREC corpus and the Mandarin News corpus (less than 10% of the smaller TREC corpus). This overlap could have been removed, but it was not, to retain the full size of both corpora in the analysis. The effect of overlap will be small.

3.1 TREC Corpus (compound words)

The TREC Corpus was obtained from the full articles in the People's Daily Newspaper from 01/1991 to 12/1993 and from the Xinhua News Agency from 04/1994 to 09/1995.

The Zipf curves for the TREC compound words are shown in Figure 7. Note that the unigram curve is different from the curve for English, first with a slope less than 1 then falling rapidly after a rank of about 1,000.

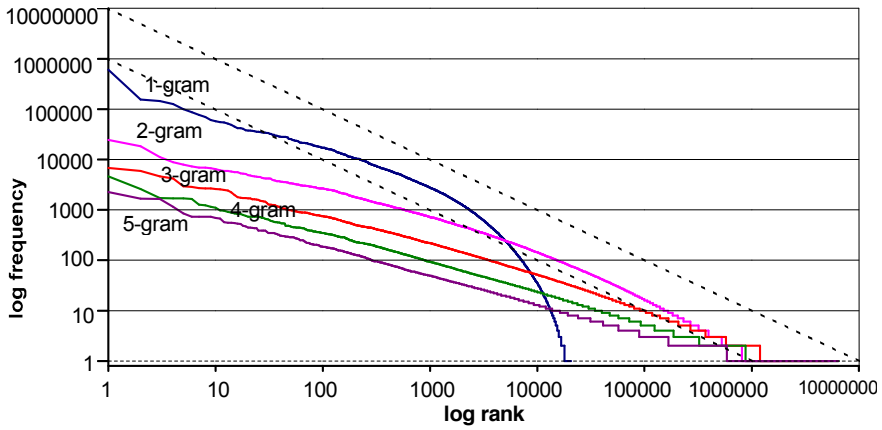

Figure 7 Zipf curves for Mandarin compound words from TREC

The crossing-point between compound word unigrams and bigrams is at rank: 4,999, and between the unigram and trigram curves at rank: 8,589, similar to English.

3.2 Mandarin News corpus (compound words)

The second corpus is the Mandarin News corpus, obtained from the People's Daily Newspaper from 1991 to 1996 (125 million syllables); from the Xinhua News Agency from

[1] http://www.ldc.upenn.edu/

1994 to 1996 (25 million syllables); and from transcripts from China Radio International broadcast from 1994 to 1996 (100 million syllables), altogether over 250 million syllables.

The Zipf curves for the Mandarin News compound words are drawn in Figure 8 and look like those for the TREC corpus. The rapid fall in the curve after rank 10,000 is due to the restricted word dictionary of 50,000 word types used in the experiment. The ten highest frequency Mandarin unigrams, bigrams and trigrams from the Mandarin News are in Table 3 and Table 4.

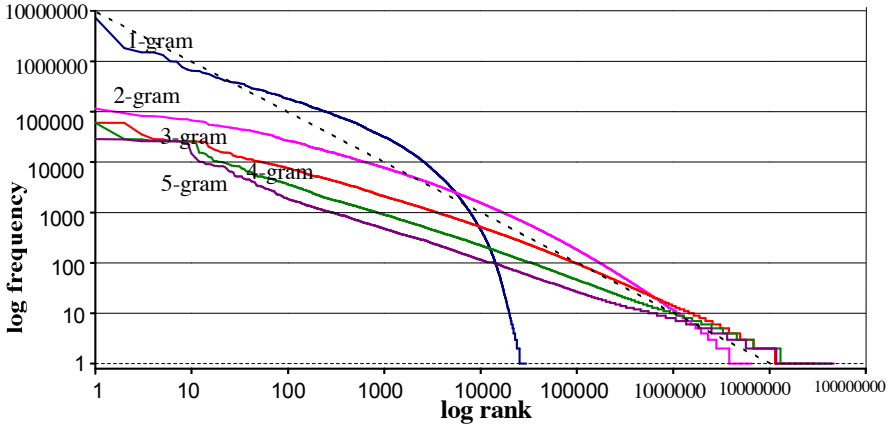

Figure 8 Zipf curves for the Mandarin News corpus (compound words)

The crossing-point between compound word unigrams and bigrams is at rank: 5,544 and between unigrams and trigrams at rank: 9,577 similar to previous values for TREC and English. So these appear to be invariants of language, not just of English.

Table 3 The ten highest frequency unigrams and bigrams from Mandarin News (compound words).

Rank	Unigrams			Bigrams		
	Freq	Token	Meaning	Freq	Token	Meaning
1	7,356,017	的	of	114,910	日 电	daily news
2	1,825,758	在	in / at	92,259	这 一	this one
3	1,515,473	和	and	82,705	这 是	this is
4	1,502,098	了	perfective marker	81,930	中国 的	of China
5	1,331,433	是	yes / right	79,390	的 发展	of development

6	989,235	一	one	75,929	他 说	he says
7	979,211	中国	China	71,922	的 一	of one
8	766,784	中	centre / middle	70,949	新 的	new of
9	686,375	有	have	70,810	日 在	daily at
10	652,004	年	year	67,211	中国 国际	China international

Table 4 The ten highest frequency trigrams from Mandarin News.

Rank	Trigrams		
	Freq	Token	Meaning
1	60,214	国际 广播 电台	international broadcast station
2	60,057	中国 国际 广播	China international broadcast
3	35,584	一 九 九	one nine nine[2]
4	28,589	据 中国 国际	according to China international
5	28,240	广播 电台 报导	broadcast station report
6	26,240	学历 收听 语言	degree listen *(to)* language
7	26,232	年龄 学历 收听	age degree listen *(to)*
8	26,203	收听 语言 备注	listen *(to)* language remarks/notes
9	26,154	职业 年龄 学历	profession age degree
10	26,081	传真 单位 职业	fax department profession

4. Zipf Curves for syllables and character strings

4.1 Chinese syllables

Because of the difficulty in extracting the compound words in Chinese, we decided to draw Zipf curves for the syllables for both Chinese corpora. TREC has 19,546,872 syllable tokens but only 6,300 syllable types, so it is not surprising that the Zipf curve for syllable unigrams

[2] This is how Chinese people read and write the year for example 1993 as "*one nine nine three*"; therefore we elminated numbers but kept the written form.

in Chinese in Figure 9 falls very rapidly after rank about 300. It is similar to previous curves, one for a smaller Chinese corpus of 2 million tokens by [Clark, Lua and McCallum 1986] and one for 10 million tokens by [Sproat 2002]. The Zipf curves for syllable *n*-grams for the TREC corpus are also shown in Figure 9.

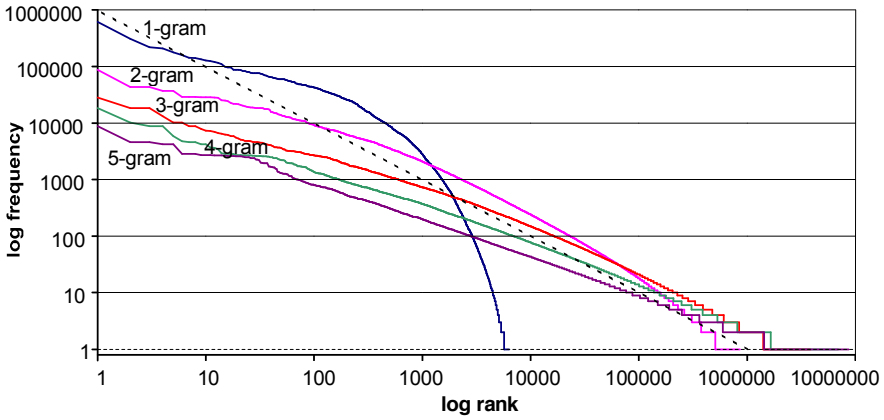

Figure 9 Zipf curves for syllables from the TREC Mandarin corpus

Except for the unigrams, the shapes of the other TREC syllable n-gram Zipf curves are similar to but not quite the same as those for compound words. In particular the syllable bigram curve for Chinese is more curved than the word curve because there are more high-frequency syllable bigrams than word bigrams. The crossing points between the syllable unigram curve and the bigram and trigram curves are at rank: 1,224 and 1,920, respectively, very different from compound words.

The number of syllable-types (i.e. unigrams) in the Mandarin News corpus is 6,800, similar to the TREC corpus. The Zipf curves and crossing points are also similar as shown in Figure 10.

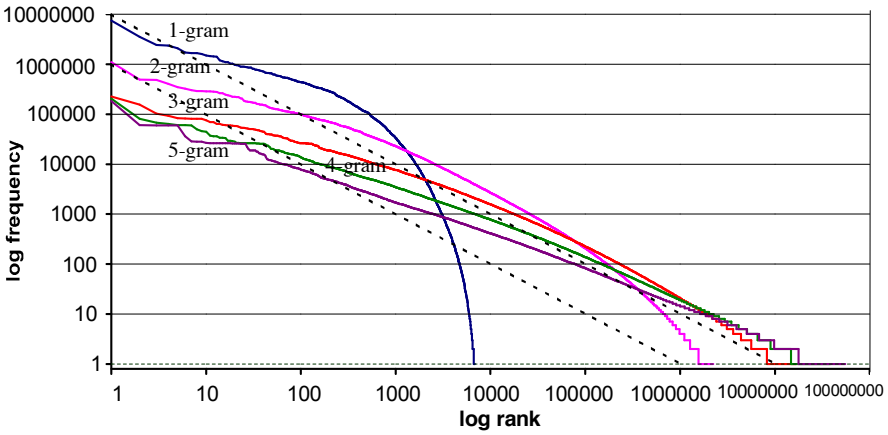

Figure 10 Zipf curves for syllables from the Mandarin News corpus

The hapax legomena and dis legomena for the Chinese corpora Zipf curves are shown in Table 5a and 5b. Their behaviour as *n* increases is similar to the English corpora.

Table 5a) Number of hapax legomena for the Chinese corpora.

Corpus		TREC syllables	TREC compound words	Mandarin News syllables	Mandarin News compound words
No of Tokens		19,720,320	13,467,443	223,222,788	153,942,010
No of Types		6,356	20,587	6,891	29,688
Hapax legomena	Unigram	676	2,642	259	4,192
	Bigram	351,691	1,013,276	667,966	2,671,406
	Trigram	2,447,451	4,009,020	8,462,775	17,794,466
	4-gram	5,309,654	5,661,530	23,812,934	30,885,192
	5-gram	7,279,824	5,875,696	37,348,300	34,617,579

Table 5b) Number of hapax dis legomena for the Chinese corpora.

Corpus		TREC syllables	TREC compound words	Mandarin News syllables	Mandarin News compound words
No of Tokens		19,720,320	13,467,443	223,222,788	153,942,010
No of Types		6,356	20,587	6,891	29,688
Hapax	Unigram	347	1,260	155	1,701

dis-	Bigram	131,619	287,225	294,634	1,001,809
legomena	Trigram	565,069	624,895	2,591,830	4,549,748
	4-gram	834,227	553,965	5,806,633	6,168,487
	5-gram	840,276	417,656	7,874,536	6,057,974

4.2 English byte substring

Following a suggestion by a reviewer of this paper, we built the Zipf curves on English 2-byte and 3-byte substrings to compare them with the Chinese syllable results.

From the WSJ88 corpus, we built a corpus of the first 2 million tokens. Then we took 2-byte and 3-byte moving windows on this corpus ignoring spaces and stopping the 2-bytes or 3-bytes at punctuation marks. As predicted by the reviewer, the results in Figure 11 and Figure 12 show that the Zipf curve for 3-byte substrings looks particularly similar to the Chinese syllable curves.

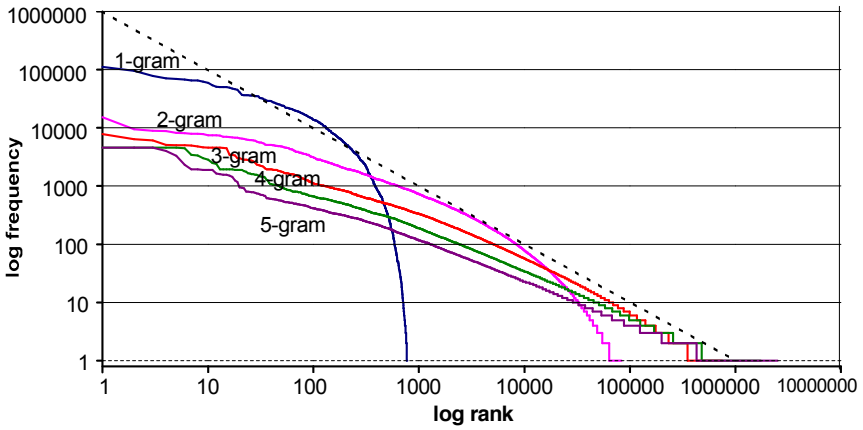

Figure 11 Zipf curves for English 2-byte substrings

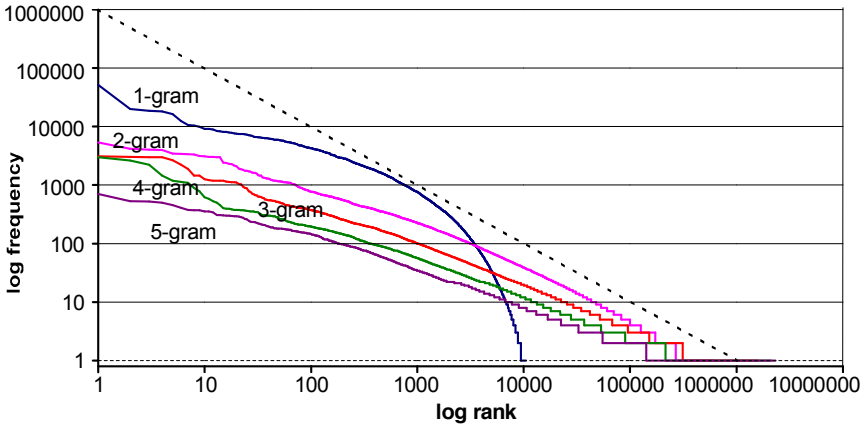

Figure 12 Zipf curves for English 3-byte substrings

Note that the number of 2-byte and 3-byte types in these curves equal 673 and 10,548, compared with the maximum possible numbers $26^2 = 676$ and $26^3 = 17{,}576$.

5. Comparison for all Zipf curves from Chinese and English

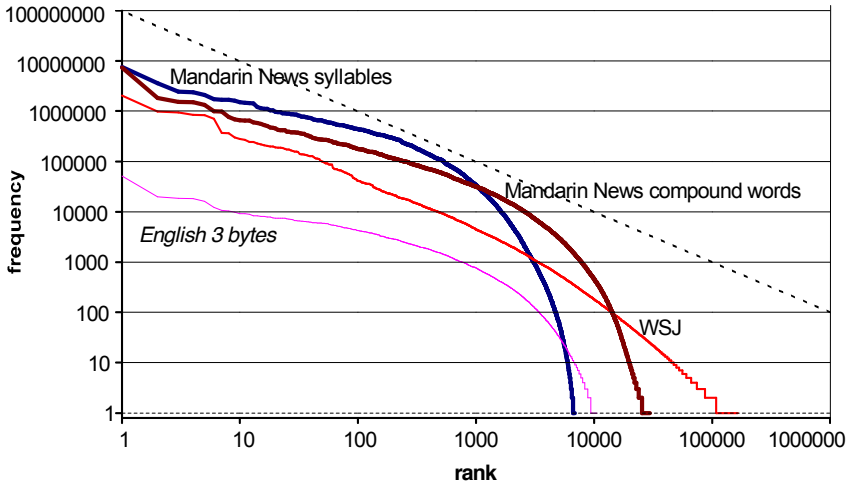

Figure 13 Comparison of Zipf curves for unigrams

The Zipf curves for unigrams for the combined WSJ corpus, the Mandarin News word corpus, the Mandarin News syllable corpus and 3-byte English corpus are compared in Figure 13.

Similarly for 2-grams to 5-grams in Figures 14 to 17.

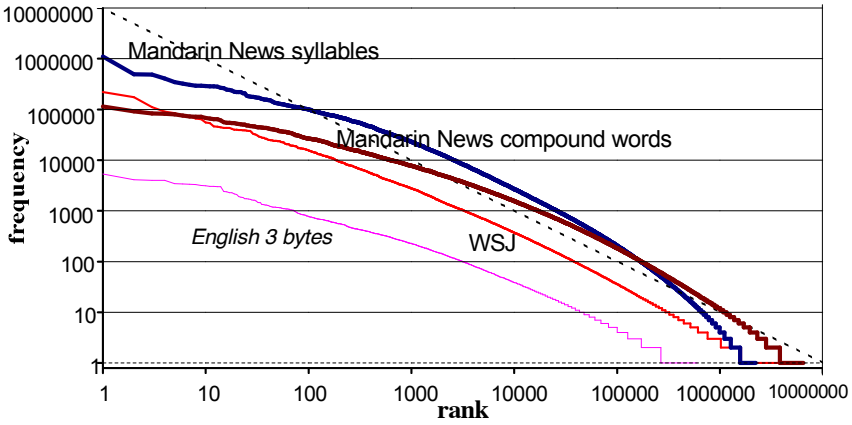

Figure 14 Comparison of Zipf curves for bigrams

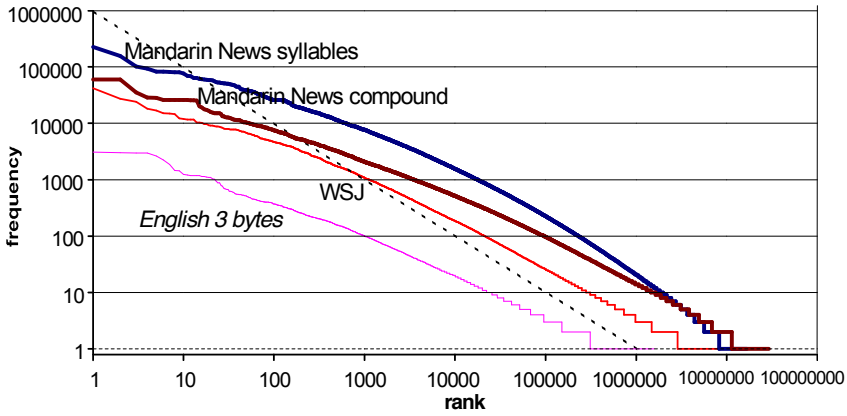

Figure 15 Comparison of Zipf curves for trigrams

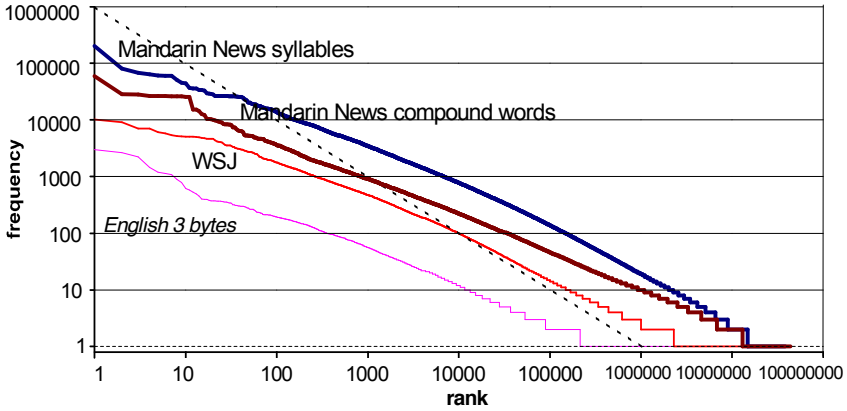

Figure 16 Comparison of Zipf curves for 4-grams

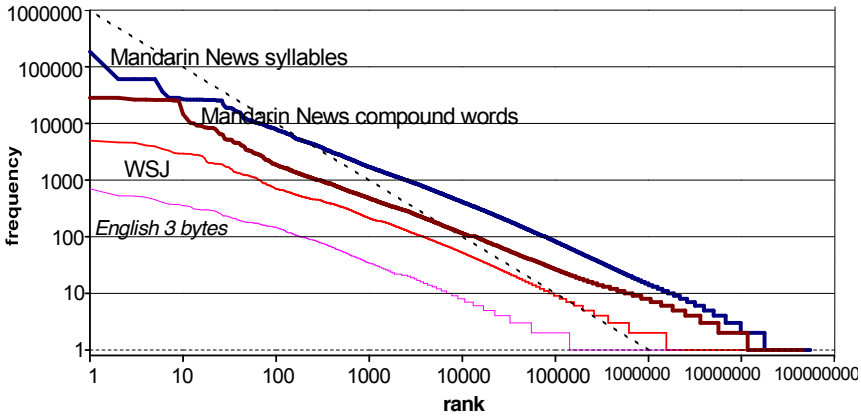

Figure 17 Comparison of Zipf curves for 5-grams

These Figures show two things as n increases. First, the curves straighten out for high n. Secondly, the number of hapax legomena becomes very large, often larger than one would expect from the last 10 steps of the rank-frequency step function. This is exactly the pattern one gets when Markov models are used to generate data sets [Baayen 1991, 2001].

6. Combined *n*-grams

The theoretical justifications for Zipf's law by Mandelbrot, Miller, Simon and others were based on single word tokens and they worked quite well for small corpora, but none of them could predict the drop in the Zipf curve below Zipf 's law for English and Chinese when the rank is greater than 5,000 word types. In the case of Chinese syllables, Zipf 's law could not hold for rank greater than about 100, but when these syllables are combined into compound words then Zipf 's law is valid for a wider range, up to about rank 1,000. Therefore, by combining Chinese syllables into larger units, Zipf 's law was extended from rank 100 to rank 1,000. This led us to combine all syllable *n*-grams, to see if the law could be extended to even higher rank and to combine word *n*-grams in Chinese and English for the same purpose.

We therefore put all unigrams and *n*-grams together with their frequencies into one large file, sorted on frequency and put in rank order as previously. The resulting combined Zipf curve is shown with the unigram curve for English words for the combined WSJ corpus in Figure 18 and for the Chinese syllables for Mandarin News in Figure 19.

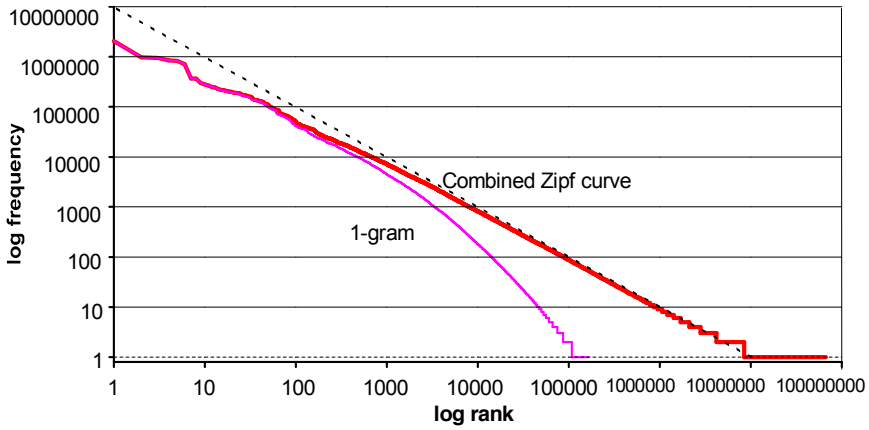

Figure 18 The unigram and combined curves for the combined WSJ corpus

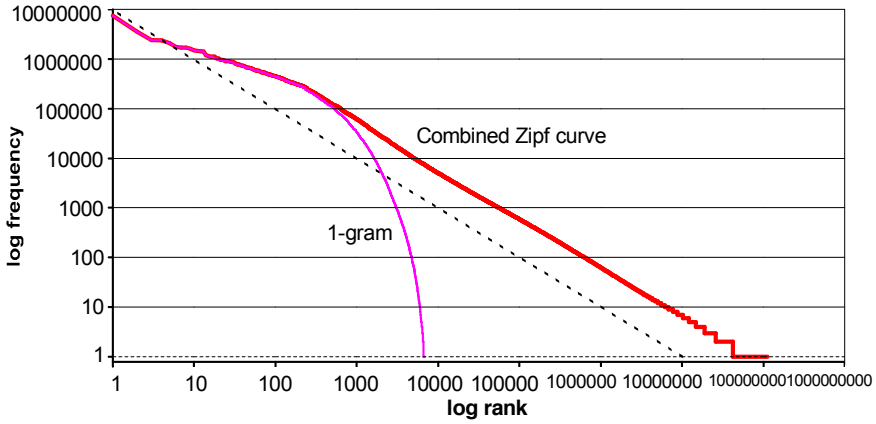

Figure 19 The unigram and combined curves for the Mandarin News syllable corpus

This shows the remarkable result that as the unigram curve drops away from Zipf 's slope of -1, the shortfall is made up almost exactly by the *n*-grams in both cases, even though those shortfalls are very different in the two cases. So when all *n*-grams are combined together, including unigrams, Zipf's law is found to be approximately correct with a slope close to -1 for all ranks. If semantically incomplete *n*-grams had been excluded from this analysis, this result would not have been obtained.

The resulting Zipf curves for the combined *n*-grams from all of the corpora are shown in Figure 20.

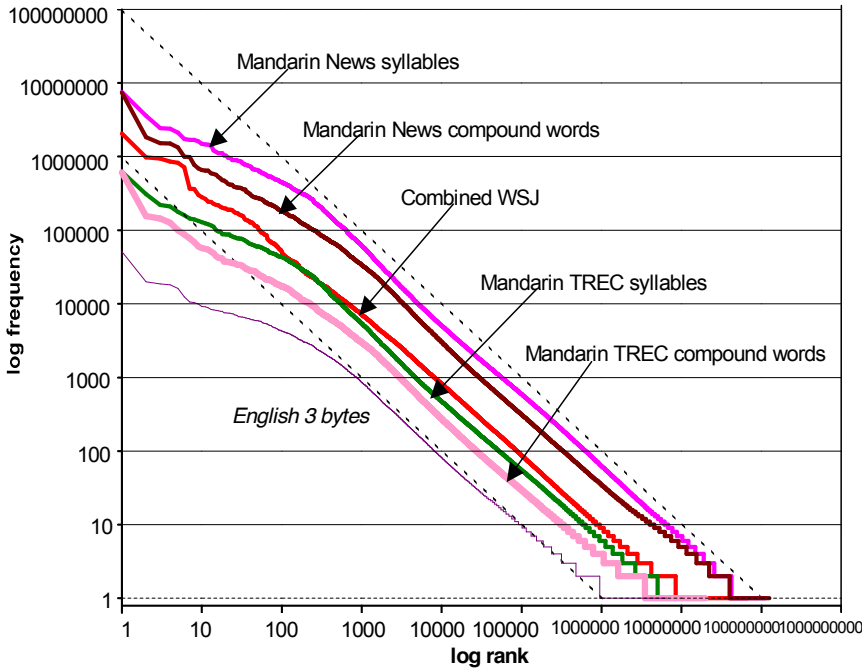

Figure 20 Combined Zipf curves for both of the languages

This shows that the 6 combined Zipf curves are all approximately straight lines with slopes close to –1 for all ranks > 1,000. For ranks < 1,000, the unigram curves dominate and are not so straight. As in Figure 18 and 19, the *n*-grams ($n \geq 2$) almost exactly make up for the deviation of the unigram Zipf curve from Zipf 's law for the six very different unigram curves. So the results in Figure 20 are a new confirmation of Zipf's original law in an extended form.

7. Summary and Conclusions

This paper reports on the results of some experiments conducted on Zipf curves for English and Chinese corpora. It was confirmed that Zipf curves on a log-log graph for single word unigram distributions for both languages fall below the straight line with slope -1 as predicted by Zipf 's law. The deviation from Zipf 's law occurs at a rank close to rank = 5,000, for the 3 corpora in English and 2 corpora in Chinese. This rank (5,000) is also the rank near which the unigram and bigram Zipf curves cross for all 5 corpora.

The more significant result was the discovery that when the frequency distribution of

words is combined with the distributions of all 2-, 3-, 4- and 5-grams, the combined Zipf curve approximately obeys Zipf 's law for all ranks and frequencies for both languages. This effectively extends Zipf 's law, with the higher *n*-grams almost exactly making up for the fall-off in the Zipf curve for words. Furthermore, this extended form of Zipf 's law also holds for the syllables of Chinese (as well as for 2-byte and 3-byte word fragments in English), even though the distribution of syllable unigrams is very different from the distribution for words.

This paper does not explain why Zipf 's law in an extended form is valid for large corpora or what this result means. This must be left for further experiments and other researchers. However, preliminary results, not yet complete, for other languages suggest that these results are universal for all languages. We also know that they do not hold for all artificial distributions of words, because some experiments with computer generated artificial distributions did not yield an extended Zipf curve, (with a random distribution, and with Zipf distributions for words with slopes $\beta = 2$ and $\beta = 0.5$).

The earlier derivations of Zipf's law due to Mandelbrot, Miller, Simon and others fail to predict the fall-off in the Zipf curve from about rank 5,000 and to predict the extended form of Zipf's law for the combined *n*-gram curves. We believe that this is because these derivations do not properly take account of the fact that each token is part of a sequence and its information is dependent on a conditional probability, conditional on the words or characters around it; this can be approximated in terms of the frequency of *n*-grams [O 'Boyle, Owens and Smith 1994].

Acknowledgement

The authors would like to express their appreciation to reviewers of this paper whose comments and suggestions made a great improvement to the paper and to Dr Xiaoyu Qiao for her contribution of testing and standardising the Chinese morphology.

References

Baayen, H. "A Stochastic Process for Word Frequency Distributions", In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL-29)*, Berkeley, California, USA, 1991, pp. 271-278.

Baayen, H. "Word Frequency Distributions", Kluwer Academic Publishers, 2001.

Bates, E., Chen, S., Li, P., Opie, M. and Tzeng, O. "Where is the boundary between compounds and phrases in Chinese ? A reply to Zhou et al.", *Brain and Language*, 45, 1993, pp. 94-107.

Booth, A. D. “A Law of Occurrences for Words of Low Frequency”, *Information and Control*, Vol. 10, No. 4, April 1967, pp. 386-393.

Carroll, J. B. "A Rationale for an Asymptotic Lognormal Form of Word Frequency Distributions", *Research Bulletin -- Educational Testing Service*, Princeton, November 1969.

Clark, J. L., Lua, K. T. and McCallum, J. (1986). "Using Zipf's Law to Analyse the Rank Frequency Distribution of Elements in Chinese Text", In *Proceedings of International Conference on Chinese Computing*, Singapore, August 1986, pp. 321-324.

Chen, S., and Shi, D-X, "On the feeding relation between syntax and morphology: Evidence from Chinese V-N compounds", In *Proceedings of the Third International Symposium on Chinese Languages and Linguistics*, Taiwan: Chinghwa University, 1992.

Fedorowicz, J. "A Zipfian Model of an Automatic Bibliographic System: an Application to MEDLINE", *Journal of American Society of Information Science*, Vol. 33, 1982, pp. 223-232.

Ferrer i Cancho, R., Solé, R. V., "Two Regimes in the Frequency of Words and the Origin of Complex Lexicons", *Journal of Quantitative Linguistics,* Vol. 8, No. 3, 2002, pp. 165 - 173.

Francis, W. N. and Kucera, H. "Manual of Information to Accompany A Standard Corpus of Present-Day Edited American English, for use with Digital Computers", Department of Linguistics, Brown University, Providence, Rhode Island, 1964.

Guiter, H. and Arapov, M., editors. "Studies on Zipf 's Law", Brochmeyer, Bochum, 1982.

Hatzigeorgiu, N., Mikros, G., and Carayannis, G., "Word Length, Word Frequencies and Zipf's Law in the Greek Language", *Journal of Quantitative Linguistics*, Vol. 8, No. 3, 2001, pp. 175 - 185.

Jelinek, F., Mercer, R. L. "Probability distribution estimation from sparse data", *IBM Technical Disclosure Bulletin*, Vol. 28, No. 6, November 1985.

Li, W. "Zipf's Law in Importance of Genes for Cancer Classification Using Microarray Data", Laboratory of Statistical Genetics, Rockefeller University, New York, 2001.

Mandelbrot, B. "An Information Theory of the Statistical Structure of Language", *Communication Theory*, edited by Willis Jackson, New York: Academic Press, 1953, pp. 486-502.

Mandelbrot, B. "Simple Games of Strategy Occurring in Communication through Natural Languages", *Transactions of the IRE Professional Group on Information Theory*, Vol. 3, 1954, pp. 124-137.

Mandelbrot, B. "A note on a class of skew distribution function analysis and critique of a paper by H. A. Simon", *Information and Control*, Vol. 2, 1959, pp. 90-99.

Mandelbrot, B. "Final note on a class of skew distribution functions: analysis and critique of a model due to H. A. Simon", *Information and Control*, Vol. 4, 1961, pp. 198-216.

Mandelbrot, B. B. "Post Scriptum to 'final note'", *Information and Control*, Vol. 4, 1961, pp. 300-304.

Miller, G. A. "Communication", *Annual Review of Psychology*, 5, 1954, pp. 401-420.

Miller, G. A. "Some effects of intermittent silence", *The American Journal of Psychology*, 52, 1957, pp. 311-314.

Miller, G. A., Newman, E. B. and Friedman, E. A. "Length-Frequency Statistics for Written English", *Information and control*, Vol. 1, 1958, pp. 370-389.

Montemurro, M. "Beyond the Zipf-Mandelbrot Law in Quantitative Linguistics", *Physica A*: *Statistical Mechanics and its Applications*, Vol. 300, Issues 3-4, November 2001, pp. 567-578.

Ney, H. "The Use of the Maximum Likelihood Criterion in Language Modelling", In *K. Ponting (*ed.): Computational Models of Speech Pattern Processing*, Springer, Berlin, Germany, 1999, pp. 259-279.

O 'Boyle, P., Owens, M. and Smith, F. J. "A weighted average *n*-gram model of natural language", *Computer Speech and Language*, Vol. 8, 1994, pp. 337-349.

Orlov, J. K. and Chitashvili, R. Y. "Generalized Z-distribution generating the well-known 'rank-distributions' ", *Bulletin of the Academy of Sciences, Georgia*, 110.2, 1983, pp. 269-272.

Packard, J. L., "The Morphology of Chinese A Linguistic and Cognitive Approach", Cambridge University Press, 2000, UK.

Paul, D. B. and Baker, J. M. "The Design for the Wall Street Journal-based CSR Corpus", In *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, Banff, Alberta, Canada, October 1992, pp. 899-902.

Samuelson, C. "Relating Turing's Formula and Zipf's Law", In *Proceedings of the 4th Workshop on Very Large Corpora*, Copenhagen, Denmark, 1996.

Sichel, H. S. "On a Distribution Law for Word Frequencies", *Journal of the American Statistical Association*, 70, 1975, pp. 542-547.

Sichel, H. S. "Word Frequency Distributions and Type-Token Characteristics", *Mathematical Scientist*, 11, 1986, pp. 45-72.

Sichel, H. S. "Modelling Species-Abundance Frequencies and Species-Individual Functions with the Generalized Inverse Gaussion-Poisson Distribution", *South African Statistical Journal*, 31, 1997, pp. 13-37.

Silagadze, Z. K. "Citations and the Zipf-Mandelbrot Law", *Complex Systems*, Vol. 11, No. 6, 1997, pp. 487-499.

Simon, H. A. "On a Class of Skew Distribution Functions", *Biometrika*, Vol. 42, 1955, pp. 425-440.

Simon, H. A. "Some Further Notes on a Class of Skew Distribution Functions", *Information and Control*, Vol. 3, 1960, pp. 80-88.

Simon, H. A. "Reply to 'final note' by Benoit Mandelbrot", *Information and Control*, Vol. 4, 1961, pp. 217-223.

Simon, H. A. "Reply to Dr. Mandelbrot's post Scriptum", *Information and Control*, Vol. 4, 1961, pp. 305-308.

Smith, F. J. and Devine, K. "Storing and Retrieving Word Phrases", *Information Processing and Management*, Vol. 21, No. 3, 1985, pp. 215-224.

Sproat, R., "Corpus-Based methods in Chinese Morphology", *Tutorial of the 19th International Conference on Computational Linguistics (COLING)*, Taipei, Taiwan, August 2002.

Tsai, J-L., Hsu, W-L. "Applying an NVEF Word-Pair Identifier to the Chinese Syllable-to-Word Conversion Problem", In *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*, Taipei, Taiwan, August 2002, pp. 1016-1022.

Yonezawa, Y. and Motohasi, H. "Zipf-Scaling Description in the DNA Sequence", In *Proceedings of the 10th Workshop on Genome Informatics*, Japan, December 1999.

Zhu, D. X. "Yufa Jiangyi (Chinese Syntax)", Shanghai: The Commercial Publisher, China, 1981.

Zipf, G. K. "Human Behaviour and the Principle of Least Effort", Reading, MA: Addison-Wesley Publishing Co., 1949.