What makes a word: Learning base units in Japanese for speech recognition

Laura Mayfield Tomokiyo
Language Technology Institute
Carnegie Mellon University
4910 Forbes Avenue
Pittsburgh, PA 15213, USA
laura@cs.cmu.edu

Klaus Ries
Universität Karlsruhe
Fakultät für Informatik
Interactive Systems Laboratories
76128 Karlsruhe, Germany
kries@ira.uka.de

Abstract

We describe an automatic process for learning word units in Japanese. Since the Japanese orthography has no spaces delimiting words, the first step in building a Japanese speech recognition system is to define the units that will be recognized. Our method applies a compound-finding algorithm, previously used to find word sequences in English, to learning syllable sequences in Japanese. We report that we were able not only to extract meaningful units, eliminating the need for possibly inconsistent manual segmentation, but also to decrease perplexity using this automatic procedure, which relies on a statistical, not syntactic, measure of relevance. Our algorithm also uncovers the kinds of environments that help the recognizer predict phonological alternations, which are often hidden by morphologically-motivated tokenization.

1 Introduction

What defines a word when there are no spaces in a written language? Words, as they are known in English and other western languages, are the basic units of recognition in most CSR systems, but when a language is written as a string of characters with no white space, how does one go about specifying the fundamental units that must be recognized? Mapping onto English-style words is one solution, but an artificial one, and may hide natural characteristics of Japanese that can be important in recognition. Recognizing phonemes, or short phoneme clusters, is another option, but recognition accuracy can improve when we have longer phoneme strings to work with; acoustic confusability decreases and a long word is a more useful predictor of subsequent words than a single syllable. Automatic segmenting tools eliminate an often inconsistent manual segmentation step, but are generally based on morphological analysis, which can produce units smaller than are desirable for speech recognition. Certainly, there exist words as can be looked up in a dictionary, but when a language is as heavily inflected as Japanese is, that only solves part of the problem. In this paper we describe an automatic process for learning base units in Japanese and discuss its usefulness for speech recognition.

2 The problem with Japanese

The Japanese language is written without spaces in between words. This means that before one can even start designing a recognition or translation system for Japanese the *units* that will be recognized, or translated, must be defined. Many sequences of phonemes, particularly those representing nouns, are clearly independent and can be designated as free-standing units. Japanese has a rich and fusional inflectional system, though, and delimiting where a verb ending ends and another begins, for example, is seldom straightforward.

Japanese has typically been segmented in variations on four ways for the purposes of recognition and parsing, although since many papers on Japanese recognition do not specify what units they are using, or how they arrived at the definition of a "word" in Japanese, it is hard to compare systems.

- Phrase/Bunsetsu level: (Early ASURA (Morimoto et al. , 1993), QJP (Kameda, 1996))
  - advantages: long enough for accurate recognition, captures common patterns
  - disadvantages: requires dictionary entry for each possible phrase; vocabulary explosion
- "Word" level: (JANUS (Schultz and Koll,

Laura Mayfield Tomokiyo and Klaus Ries (1997) **What makes a word: Learning base units in Japanese for speech recognition.** In T.M. Ellison (ed.) *CoNLL97: Computational Natural Language Learning,* ACL pp 60-69.

1997))

  - advantages: units long enough not to cause confusion, but short enough to capture generalizations
  - disadvantages: not natural for Japanese; easy to be inconsistent; may hide qualities of Japanese that could help in recognition

- Morpheme level: (Verbmobil (Yoshimoto and Nanz, 1996))
  - advantages: mid-length units that are natural to Japanese
  - disadvantages: a lot of room for inconsistency; "morpheme" can be interpreted broadly and if segmented in the strictest sense units can be single phonemes

- Phoneme cluster level: (NEC demi-syllable (Shinoda and Watanabe, 1996)), JANUS KSST[1]
  - advantages: only need a short dictionary
  - disadvantages: high confusability, although confusability seems less of a problem for Japanese than some other languages

The bunsetsu is a unit used to segment Japanese which generally consists of a content component on the left side and a function component on the right side. Bunsetsu boundaries seem to be natural points for pausing and repetition, and most elementary schools include bunsetsu segmentation as a formalized part of grammar education. *John-ga* ("John-NOM"), *hon-o* ("book-ACC"), and *yonda* ("gave") are all examples of bunsetsu.

Bunsetsu can be quite long in terms of both phonemes and morphemes, however, and quite unique. For example, *saseteitadakitaindesuga* would be considered a single bunsetsu. This phrase contains a causative form of the verb "to do", *sase-*, a gerunditive suffix *-te-*, the root of a formal verb meaning to receive *-itadaki-*, a desidirative suffix *-tai-*, a complementizer *-n-*, a copula *-desu-*, and a softener *-ga*.

3 Our approach

Our approach, described in detail in (Ries et al., 1996), uses a statistical tool that automatically finds important sequences. This tool was originally developed to help mitigate the bias introduced by a word-based orthography by explicitly modeling important multi-word units. The target of the tool was languages for which the word seemed already a useful level of abstraction from which to expand, and experiments were first performed on English and German for the scheduling task. One important motivation for this work was the desire to capture lexicalized expressions that exhibit, in natural speech, markedly different pronunciation from what concatenating the constituent words would predict. Examples of such expressions are *don't-know* (dunno), *i-would-have* (ida), *you-all* (yaw).

The objective of the phrase-finding procedure is to find a pair of frequently co-occuring basic units for which joining all occurrences in the corpus is a useful operation. Until very recently most implementations of this idea have made use of measures of co-occurrence that have been useful in other domains, and the pair is chosen by maximizing that criterion. In contrast we assume that we want to model the corpus with a statistical language model and search for those sequences that increase the modeling power of the model by the largest amount. Our measurements are based on information theoretic principles and the usage of m-gram models of language, a common practice in the speech community. The model described here will therefore implicitly consider the words surrounding the phrase candidates and use information about the context to determine the goodness of a sequence, which is in contrast to traditional measures.

(Ries et al., 1996) has compared a variety of measure as reported in the literature and has found these to be not competitive with the new technique if used in statistical language models. In a very vague statement we want to add that this corresponds to the experience in eyeballing these sequences. The measures that were compared against in this earlier work have been:

- mutual information (Magerman and Marcus,)
- frequency
- iterative marking frequency (Ries et al., 1995)
- backward bigram: $p(\mathrm{w}_1|\mathrm{w}_2)$
- backward perplexity: $p(\mathrm{w}_1, \mathrm{w}_2) \cdot \log(p(\mathrm{w}_1|\mathrm{w}_2))$
- Suhotin's measure (Suhotin, 1973)

3.1 Statistical language modeling and speech recognition

Statistical models of language are, to our knowledge, the type of language model used in all modern speech

[1]Korean Spontaneous Scheduling Task; SST described more fully in Section 4.1

recognition engines, especially in research systems but also in most commercial large vocabulary systems that can recognize naturally fluent spoken language. In principle the speech recognition problem is to find the most likely word sequence $\mathbf{W}$ given the acoustic $\mathbf{A}$.

$$\mathrm{argmax}_{\mathbf{W}} p(\mathbf{W}|\mathbf{A})$$

Using Bayes theorem and the knowledge that $p(\mathbf{A})$ does not change the maximization we arrive at

$$\mathrm{argmax}_{\mathbf{W}} p(\mathbf{A}|\mathbf{W}) \cdot p(\mathbf{W})$$

$p(\mathbf{A}|\mathbf{W})$ is commonly referred to as the acoustic model, $p(\mathbf{W})$ is the language model and the argmax operator is realized by specialized search procedures. This paper for the most part ignores the search problem. The acoustic model is in part influenced by the sequences since we can change the entries in the pronunciation dictionary that encode the phoneme sequences the speech system uses to generate its models. During this generation process most modern systems make only partial use of neighboring words and the construction process is up to date also unable to model contractions, especially at word boundaries. It is therefore of great advantage to have a basic unit in the decoder that allows for manual or automatic dictionary modification that captures these phenomena. This has recently been reported to be a very promising modeling idea on several different speech recognition tasks in English. The underlying assumption is that sequences of units that have a high stickiness are by conventional usage very likely to show idiosyncratic pronuncations much like single words do: They are for the most part lexicalized.

The statistical language modeling problem for the sequence of words $\mathbf{W} = w_1, \ldots, w_n$ where w_n is a special end of sentence symbol can then be rephrased as

$$p(\mathbf{W}) = \prod_{i=1}^{n} p(w_i|w_1, \ldots, w_{i-1})$$

We will for most applications probably never be able to find enough data to estimate p as presented above. An often practiced shortcut is therefore to assume that each word is only dependent on the last $m-1$ words and that this distribution is the same in all positions of the string. These models are called m-gram models and have proved to be very effective in a large number of applications, even though they are a naive model of language.

Information theoretic measures (Cover and Thomas, 1991) are frequently used to describe the power of language models. (Cover and Thomas, 1991) shows in chapter 4.2 that the entropy rate of a random process converges, under additional assumptions, to the entropy of the random source. This has been taken as the justification for using an approximation of a notational difference of the entropy rate,dubbed *perplexity*, as a measure of the strength of the language model. Given a bigram model p and a test text $w_1, \ldots, w_n$ the perplexity PP is defined as

$$PP = 2^{-\frac{1}{n}\sum_{i=1}^{n} \log P(w_i|w_{i-1})}$$

where we make usage of a special "start-of-sentence" symbol as w_0. In the sequel we happily ignore this for notational convenience.

Since we will be changing the basic unit during the sequence finding procedure it is useful to normalize the perplexity onto one standard corpus. Say the standard test corpus has length n and the new test corpus has length n' we define for the test corpus $PP^{rel} = PP^{\frac{n'}{n}}$. PP^{rel} is therefore a notational variant of the probability of the test text given the model which is independent of the used sequences of words and is the only meaningful measure in this context.

The calculation of the model p itself from empirical data involves a number of estimation problems. We are using the well understood and empirically tested backoff method, as recently described e.g. by (Kneser and Ney, 1995).

3.2 Algorithm description

The idea of the algorithm is to search for sequences that reduce the relative perplexity of the corpus in an optimal way. For example, if we were working with a bigram model and came across the sequence *credit card bill*, not only would we have to choose among words like "report," "history" and "check" as possible successors for "credit," but the word "card" itself has many senses and "game," "shop" and "table" might all be more likely followers of "card" than "bill," if no other context is known. By creating a new word, *credit_card*, we eliminate one choice and decrease the surprise of seeing the next word.

Since the new word is now treated exactly like other word instances in the corpus, it can in turn be the first or second half of a future joining operation, leading to multi-word compounds.

The sequence-finding algorithm iterates over all word pairs in a training corpus, and in each iteration chooses the pair (recall that one or both elements of this pair can themselves be sequences) that reduces the bigram perplexity the most. This can be done by just calculating the number of times all possible word triples appeared and going over this table (except for those entries that have a count of zero) once. This is iterated until no possible compound reduces perplexity. This technique is obviously just an approximation of an algorithm that considers all word sequences at once and would allow the statistical model to produce the components of a sequence separately. The clustering is therefore a bottom up procedure and during the training of our models we are making a variation of the Viterbi assumption in joining the sequences in the corpus blindly.

For the corpora we worked with, this technique was sufficiently fast with the efficient implementation described in (Ries et al., 1996), which makes further use of estimation tools from pattern recognition such as the leaving one out technique.

Inspired by (Lauer, 1995), we have very recently extended this technique so that the algorithm has the option of, instead of replacing a sequence of two units by a new symbol, replacing it by either the left or right component of that sequence. The idea is that the resulting model could capture head information. We have tested this approach on some of our English corpora; the resulting sequences look unpromising, however, and the new option was seldom used by the algorithm.

3.3 Application to Japanese

Realizing that the phrase-finding procedure we used on English and German was producing units that were both statistically important and semantically meaningful, we decided to apply the same techniques to Japanese. We needed units that were long enough for recognition and wanted to generalize on inflected forms that are used over and over again with different stems, as well as longer sequences that are frequently repeated in the domain. Other motivations for such a process include:

- language model estimation
- preserving important cross-morphological phonetic environments
- inconsistency of human transcribers
- search sub-optimality due to poorly chosen units

The approach described in Section 3.2 is a bottom-up approach to sequence finding, and the segmentation of Japanese is more intuitively viewed as a top-down problem in which an input string is broken down to some level of granularity. In applying the algorithm in (Ries et al., 1996) to Japanese, we reversed the problem, first breaking the corpus down to the smallest possible stand-alone units in Japanese, and then building up again, constructing phrases.

We chose the mora as our fundamental unit. A mora is a suprasegmental unit similar to a syllable, with the important distinctions that a mora does not need to contain a vowel (syllabic /n/ and the first of double consonants are considered independent morae) and a mora-based segmentation would treat long vowels as two morae. The word *gakkoo* (school) would be two syllables, but four morae: *ga-k-ko-o*. Each kana of the Japanese syllabary represents one mora. In some cases kana can be combined and remain a single mora; *kyo*, as in Tokyo, is an example.

There is some argument as to whether it is natural to break multi-phoneme (CV) kana down further, to the phoneme level; specifically, some analyses of Japanese verb inflections consider the root to include the first phoneme of the alternating kana, as shown in Table 1.

kana		phoneme		example
stem	infl.	stem	infl.	
hashi	ra	hashir	a	hashiranai
hashi	ri	hashir	i	hashirimasu
hashi	ru	hashir	u	hashiru
hashi	re	hashir	e	hashireba
hashi	ro	hashir	o	hashiroo

Table 1: kana-based vs. phoneme-based analyses of verb stems and inflections

The nasal consonant kana is considered an independent unit.

The problem of segmentation is not unique to Japanese; there are other languages without spaces in the written language, and verb conjugations and other inflective forms are issues in almost any language. Words as defined by orthography can be more a curse than a blessing, as having such convenient units of abstraction at our disposal can blind us to more natural representations.

(Ito and Kohda, 1996) describes an approach similar to ours. Our work is different because of the phrase finding criterion we use, which is to maximize the predictive power of the m-gram model directly. The recent (Ries et al., 1996) showed that a variation of that measure, coined bigram perplexity, outperforms classical measures often used to find phrases. For Chinese (Law and Chan, 1995), a similar measure was combined with a tagging scheme since the basic dictionary already consisted of 80,000 words. The algorithm presented in (Ries et al., 1996) is comparatively attractive computationally, and avoids problems with initialization as it works in pure bottom up fashion. Ries did not find specific improvements from using word classes in the tasks under consideration.

Masataki (Masataki and Sagisaka, 1996) describes work on word grouping at ATR, although what they describe is critically different in that they are grouping previously defined words into sequences, not defining new words from scratch. Nobesawa presents a method for segmenting strings in (Nobesawa et al. , 1996) which uses a mutual information criterion to identify meaningful strings. They evaluate the correctness of the segmentation by cross-referencing with a dictionary, however, and seem to depend to a certain extent on grammar conventions. Moreover, a breaking-down approach is less suitable for speech recognition applications than a building-up one because the risk of producing out-of-vocabulary strings is higher. Teller and Batchelder (Teller and Batchelder, 1994) describe another segmentation algorithm which uses extensively knowledge about the type of a character (hiragana/katakana/kanji, etc). This work, though, as well as Nobesawa's, is designed for processing Japanese text, and not speech.

Our process is similar to noun compounding procedures, such as described in (Lauer, 1995), but does not use a mutual information criterion. The algorithm was originally developed to find sequences of words in English, initially in order to reduce language model perplexity, then to predict sequences that would be contracted in fast speech, again in English. The work described in this paper is an application of this algorithm to learning of word units in Japanese.

4 Evaluation

Since the phrase-finding algorithm described in 3.2 is designed to maximize bigram perplexity, the evaluations described here measure this criterion.

4.1 Task

The Spontaneous Scheduling Task (SST) databases are a collection of dialogues in which two speakers are trying to schedule a time to meet together. Speakers are given a calendar and asked to find a two-hour slot given the constraints marked on their respective calendars. Dialogues have been collected for English (ESST), German (GSST), Spanish (SSST), Korean (KSST) and Japanese (JSST).

4.2 Test corpora

Six language models were created for the scheduling task JSST (Schultz and Koll, 1997). The models were drawn from six different segmentations of the same corpus, as described below. Segments (also referred to as "chunks") were found using the compounding algorithm described in Section 3.2.

1. Corpus C1 comprised only romanized mora syllables. A romanization tool was run over the original kanji transcriptions; the romanized text was then split into kana (morae).
2. Corpus C2 was the result of running C1 through the sequencer.
3. Corpus C3 comprised chunks that were learned *before* romanization. The chunked kanji text was then run through the same romanization tool.
4. Corpus C4 was a hand-edited version of C3, where some word classes (like day of the week - if only "tuesday" existed in the corpus the rest of the days were added by hand) were fleshed out and superfluous chunks removed.
5. Corpus C5 was the hand-segmented text used in the current JSST system, with the errorful segmentations described in 5
6. Corpus C6 was C5 + chunks from C4

Only experiments involving romanized corpora were used. The choice of using romanized text over kana text was primarily based on the requirements of our language modeling toolkit; we used a one-to-one mapping between kana and roman characters. Equipped with a list of chunks (between 800 and 900 were identified in these corpora), one can always reproduce kanji representations. Breaking down a kanji-based corpus, though, would require a dictionary entry for each individual kanji, of which there are over 2500 that occur in our database. Not only is this difficult to do, given the 3-12 possible readings for each kanji, we would be left after the chunking process with singleton kanji for which it is often impossible to determine the correct reading out of context. One experiment combining chunks extracted from a kanji corpus with chunks from a kana corpus was performed, but the results were not encouraging. Kanji are an extremely informative form of representation, and we will continue to look for ways to incorporate them in future work. However, experiments do show that even without them phrase-building can produce significant results.

4.3 Perplexity results

The relative perplexities reported below are all normalized with respect to corpus C1. The result below clearly indicates that we can do at least as good or even better than human segmentations using automatically derived segmentations from the easily definable mora level. We also want to point out that the sequence trigram is better than a four-gram which indicates that the sequences play a critical role in the calculation of the model.

Our measure of success so far is relative perplexity, and for speech recognition the ultimate measure is of course the accuracy of the recognition results. These results however are in our judgement much better than our results on English or German and we are hopeful that we can integrate this into our JANUS - Japanese speech system.

	PP^{rel}	corpus size	vocab size
mora			
C1	6.1	38963	189
C1 4-gram	4.7	39995	189
C2	4.5	16070	1058
kanji chunks			
C3	4.7	19400	1118
hand-edit			
C4	4.6	19135	977
"words"			
C5	6.3	25951	2357
C6	6.0	25575	3286

The dictionary size is the base dictionary size, without the chunks included. The mora dictionary has only 189 word types because it comprises only the legal syllables in Japanese, plus the letters of the alphabet, human and non-human noise, and some other symbols. The word dictionary, used in modeling C5 and C6, had 2357 word types.

To make the results as strong as possible we used a pseudo closed vocabulary for C5 and C6. This means that we included all word types that occur in the training and test set in the vocabulary. The dictionary size is therefore exactly the number of word types found in both training and test sets and includes the number of sequences added to the model. This favors C5 and C6 strongly, since words that are not in the dictionary cannot be predicted by the language model at all nor can a speech recognition system detect them. However this setup at least guarantees that the models built for C5 and C6 predict all words on the test set as C1-4 do. For larger tasks we assume that the unknown word problem in Japanese will be very pronounced.

A speech system can obviously recognize only words that are in its dictionary. Therefore, every unknown word causes at least one word error, typically even more since the recognizer tries to fit in another word with a pronounciation that does not fit in well. This may lead to wrong predictions of the language model and to wrong segmentations of the acoustic signal into base units. C1-C4 have a closed vocabulary that can in principle recognize all possible sentences and these segmentations do not suffer from this problem.

In English, this would be equivalent to having been able to build phoneme based language models that are better than word models, even if we choose the vocabulary such that we have just covered the training and test sets. In some pilot experiments we actually ran the sequence finding procedure on an English phoneme corpus and a letter corpus without word boundaries and found that the algorithm tends to discover short words and syllables; however, the resulting models are not nearly as strong as word models.

5 Emergence of units

One of the exciting things about this study was the emergence of units that are contracted in fast and casual speech. A problem with morphological breakdowns of Japanese, which are good for the purposes of speech recognition because they are consistent and publicly available tokenizers can be used, is that multi-morph units are often abbreviated in casual speech (as in "don't know" ⇒ "dunno" in English) and segmenting purely along morphological boundaries hides the environment necessary to capture these phenomena of spontaneous speech. We found that the chunking process actually appeared to be extracting these sequences.

5.1 Reducible sequences captured

Following is an example comparing the chunking to the original (termed word-based here) segmentation in JSST. The task, again, is appointment scheduling. Numbered sentences are glossed in Table 2; (1) and (6) correspond to (A); (2,7) to (B); (3,8) to (C), etc.

(1) gozenchuu ni shi te itadake reba
(2) getsuyoobi ni shi te itadakere ba to omoi masu
(3) ukagawa shi te itadakere ba
(4) renraku shi nakere ba to omot te
(5) sorosoro kime nake re ba nara nai

Sentences 1-5 are shown as segmented by human transcribers. Sentences 6-10 are the same three sentences, segmented by our automated process.

(6) ⟨gozenchuu⟩ ni ⟨shiteitada⟩ ⟨kereba⟩
(7) ⟨getsuyoobi⟩ ni ⟨shiteitada⟩ ⟨kereba⟩ ⟨toomoimasu⟩
(8) ⟨ukagawa⟩ ⟨shiteitada⟩ ⟨kereba⟩

(A)	*gozenchuu-ni* $	*shite*	*itadakereba* $			
	in the morning	do	if I could receive the favor of			
	If you would be so kind as to make it in the morning . . .					
(B)	*getsuyoobi-ni* $	*shite*	*itadakereba-to* $			*omoimasu* $
	on monday	do	if I could receive the favor of-COMP			[I] think
	If you would be so kind as to make it on monday . . .					
(C)	*ukagawashite* $		*itadakereba* $			
	cause to humbly go		if I could receive the favor of			
	If you would allow me to go . . .					
(D)	*renraku*	*shinakereba*	*to* $	*omotte*		
	contact	if [I] don't	COMP	thinking		
	I've been meaning to get in touch [with you/him. . .]					
(E)	*sorosoro* $	*kimenakereba*	*naranai* $			
	soon	if [I] don't decide	it won't do			
	[I] have to decide soon . . .					
(F)	*nan*	*tte-yuu-ka* $				
	what	COMP-say-QUE				
	what to say . . .					
(G)	*sono-hi-wa* $	*gogo-wa* $	*muri-desu* $	*to-yuu-ka* $		*sanji-made* $
	that-day-TOP	afternoon-TOP	impossible-COP	COMP-say-QUE		until-three
	kaigi-ga $	*haitte-iru-node* $	*sanji-ikoo-nara* $	*daijoubu-desu-kedo* $		
	meeting-SUBJ	in-is-because	three-after-if	okay-COP-SOFTENER		
	That afternoon is impossible - that is to say, there's a meeting until three, so if it's after three it would be okay					
(H)	*asa*	*hayaku-to* $	*yuugata-nara* $	*aite*	*(i)masu*	*kedo* $
	morning	early-and	evening-if	open	is	SOFTENER
	early morning and evening are open					
(J)	*sanji*	*made* $	*kaigi*	*ga* $	*haitte*	*orimasu* $
	3:00	until	meeting	SUBJ	in	is
	[There] is a meeting until 3:00					

Table 2: Glosses of sentences (1) through (17). Space boundaries vary to illustrate the specific issues being discussed at the point in the text where the sentences occur; dollar signs indicate bunsetsu boundaries.

(9) ⟨renraku⟩ shi na ⟨kereba⟩ ⟨toomo⟩ ⟨tte⟩

(10) ⟨sorosoro⟩ ⟨kime⟩ na ⟨kereba⟩ ⟨nara⟩ ⟨nai⟩

There are two issues of importance here. First, the hand-segmenting, while it can be tailored to the task, is inconsistent; the sequence ". . .ni-shi-te-i-ta-da-ke-re-ba" (If I could humbly receive the favor of doing...) is segmented at one mora boundary in (1) and at another in (2). Sentences (4) and (7) show the same sequences as segmented by the chunker; the segmentation is consistent. The same is true for ". . .na-ke-re-ba in (4) and (5) as compared to (9) and (10).

The second important issue is the composition of the sequences. The sequence "kereba" in (6-10), while used here in a formal context, is one that is often reduced to "kya" or "kerya" in casual speech. The knowledge that "kereba" can be a word is very valuable for the speech recognizer. Once it has access to this information, it can train its expected pronunciations of the sequence "kereba" to include "kya" pronunciations as they occur in the spoken corpus. Without the knowledge that these three morae can form one semantic unit, the recognizer cannot abstract the information that when combined in certain contexts they can be reduced in this special way.

Although the ⟨kereba⟩ in (6) and (7) is attached to a verb, *itadaku*, that is very formal and would not be abbreviated in this way, let us consider sentences (D) and (E), here segmented into bunsetsu phrases:

(11) renraku shinakereba to omotte

(12) renraku shinakya to omotte

(13) sorosoro kimenakereba naranai

(14) sorosoro kimenakya naranai

Sentence (D) is shown in (11) in full form and in (12) in contracted form; sentence (E) is shown in (13) in full form and in (14) in contracted form. Selection of the chunk ⟨kereba⟩ provides the environment necessary for modeling the contraction "kya" with some verbs and adjectives in informal speech.

Basing a tokenizer on syntactic factors can hide precisely such environments.

A second example of a frequently contracted sequence in Japanese is *to yuu* or *tte yuu* which becomes something close to “chuu” or “tyuu” in fast and sloppy speech.

(15) naN tte yuu ka

(16) sono hi wa gogo wa muri desu, to yuu ka, sanji made kaigi ga haitte iru node sanji ikoo nara daijoubu desu kedo

The *to yuu* sequence is recognized as a single sequence in some tokenization methods and not in others, so the idea of treating it as a single word is not novel, but in order for the variant “chuu” to be considered during recognition, it is important that our system recognize this environment.

There are cases in which the combination *to yuu* will not collapse to “chuu:”

(17) asa hayaku to yuugata nara aitemasu kedo

In the scheduling domain, the word *yuugata* (evening) is common enough for it to be identified as a word on its own, and the utterance is correctly segmented as ⟨to⟩ ⟨yuugata⟩. In a different domain, however, the extraction of ⟨toyuu⟩ might take precedence over other segmentation, which would indeed be incorrect.

Yet another type of contraction common in casual speech is blending of the participial suffix *te* and the beginning of the auxiliary *oru*, as in (J).

The -te form of the verb, also often referred to as the participial (Shibatani, 1987) or gerundive (Matsumoto, 1990) form, is constructed by adding the suffix *te* to the verb stem plus the *renyoo* inflection. This *renyoo* (conjunctive) form of the verb is also used with the past-tense suffix *ta* and provisional suffix *tara*.

In the majority of the literature, the -te form seems to be analyzed either as a single unit independent of the auxiliary verb (iru/oku/aru/morau etc.)(Sells, 1990) or broken down into its morphological constituents (Yoshimoto and Nanz, 1996). An exception is (Sumita and Iida, 1995). With certain auxiliary verbs, though, the *e* in *te* is dropped and the suffix-initial *t* is affixed to the initial vowel of the auxiliary, as in *hait-torimasu, shi-tokimasu.* This phenomenon is very pronounced in some dialects and only slight in others.

Our method does identify several units that have the *-te* appended directly onto the auxiliary verb, creating a very useful phonetic environment for us.

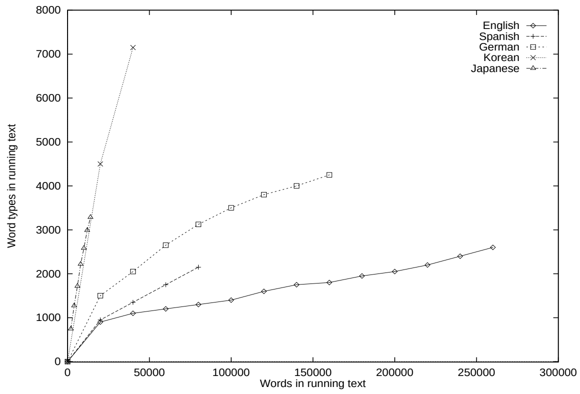

Figure 1: Vocabulary growth rates for English, Spanish, German and Korean for the Spontaneous Scheduling Task (SST).

5.2 Long enough for speech recognition

In speech recognition systems, short recognition units are to be avoided because they are confusible - it is much harder to distinguish between ”bee” and ”key” than ”BMW” and ”key lime pie.” This is one reason that we did not want to use a morphological breakdown of input sentences. Segmented in the strictest sense (Teller and Batchelder, 1994), the sentence “[I] was studying” could be written as:

benkyoo	*shi*	*te*	*i*	*mashi*	*ta*
study	do	PART	PROG	POLITE	PAST

Single-phoneme units like /i/ and syllabic /n/ are so small that they are easy to misrecognize. Even /te/ and /ta/ are shorter than would normally be desired, although Japanese phonemes appear to be less confusible than their English and German counterparts (Schultz and Koll, 1997). Units such as ⟨shite⟩ and ⟨imashita⟩, as produced by our algorithm, are long enough to be distinguishable from other words, yet short enough to generalize. Since the basic unit from which we were building was the mora, ending up with units that were too short was a concern. We found that the average unit length in mora was comparable to that of the hand-segmented system, however.

It is also important, though, to control the vocabulary size if a reasonable search space is desired. Early experiments with recognizing at the bunsetsu level in Korean indicated that vocabulary did explode, since most full bunsetsu were used only once. The vocabulary growth actually did level off eventually, but the initial growth was unacceptable, and we switched to a syllable-based system in the end. Figure 5.2 shows vocabulary growth rates in Janus for different languages in the scheduling domain.

5.3 Undesired effects

Since our algorithm evaluates all sequences with the same components identically, some compounding that is clearly wrong occurs.

5.3.1 Component sharing

For example, the chunk ⟨kuno⟩ was identified by the system. This was because the phrases *daigaku-no* "university-GEN" and *boku-no* "I/me-GEN" were both very common - the algorithm abstracted incorrectly that ⟨kuno⟩ was a meaningful unit before it found the word *daigaku*, which it eventually did identify.

5.3.2 Incomplete sequences

Although the point where a stem should end and an inflection begin can be ambiguous, most stems have definite starting points, and this algorithm can miss them. For example, *mooshiwakegozaimasen* "I'm very sorry" occurs many times in the database, but our algorithm only extracted part: ⟨shiwakegozaimaseN⟩. Because of the way our stopping criterion is defined, we can infer from the fact that the full phrase was not extracted that by forming this compound we would actually have *in*creased the difficulty of the corpus; more analysis is needed to understand exactly why.

6 Conclusion

The results reported here show that we can get similar entropies in our language model by using an automatic process to segment the data. This means that we do not have to rely on human segmenters, which can be inconsistent and time consuming. We can also tailor the segmentation style to the task; the inflected forms and general word choice in casual and formal speech are very different, and our method allows us to target those which are most relevant. This is in itself a significant result.

Additionally, we found that our method finds sequences which are likely to undergo contractions and reductions in casual speech. This has implications not only for Japanese, but also for speech recognition in general. If our algorithm is finding a natural base unit in Japanese, we should be able to use a similar approach to find units more natural than the word in other languages.

7 Acknowlegdements

This research was performed at the University of Karlsruhe and at Carnegie Mellon University, Pittsburgh. The authors were supported by project VerbMobil through the German BMBF. We gratefully acknowledge their support. The researchers also would like to thank Professor Kurematsu of the University of Electro-communications in Japan for providing the environment for this research as well as valuable advice. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of any organization mentioned above.

References

Thomas M. Cover and Joy A. Thomas Elements of Information Theory. Wiley, 1991. Series in Telecommunications.

Sabine Deligne and Frederic Bimbot. Language Modeling by Variable Length Sequences: Theoretical Formulation and Evaluation of Multigram. In *ICASSP* 1995, Vol. 1, pp. 169-172.

Akinori Ito and Masaki Kohda. Language Modeling by String Pattern N-gram for Japanese Speech Recogniton. In *ICSLP*, 1996.

Masayuki Kameda. A Portable & Quick Japanese Parser: QJP. In *COLING*, Copenhagen, 1996.

Reinhard Kneser and Hermann Ney. Improved Backing-off for M-gram Language Modeling. In *ICASSP* 1995, Vol. 1, pp. 181-184.

Mark Lauer. Corpus Statistics Meet the Noun Compound: Some Empirical Results. In *ACL*, 1995.

Hubert Hin-Cheung Law and Chorkin Chan. Ergodic Multigram HMM Integrating Word Segmentation and Class Tagging for Chinese Language Modeling. In *ICASSP* 1996, Vol.1, pp. 196-199.

David M. Magerman and Mitchell P. Marcus. Distituent Parsing and Grammar Induction. pages 122a–122e.

Sven Martin, Joerg Liebermann, and Hermann Ney. Algorithms for Bigram and Trigram Clustering. In *Eurospeech*, 1995.

Hirokazu Masataki and Yoshinori Sagisaka. Variable-order N-gram Generationi by Word-class Splitting and Consecutive Word Grouping. In *ICASSP* 1996, Vol. 1, pp. 188-191.

Yo Matsumoto. Constraints on the 'Intransitivizing' Resultatitive -te aru construction in Japanese.

In *Japanese/Korean Linguistics*, pp. 269-283, SLA, Stanford, 1990

Michael K McCandless and James R Glass. Empirical Acquisition of Language Models for Speech Recognition. In *ICSLP*, Yokohama, Japan, 1994.

Tsuyoshi Morimoto et al. ATR's Speech Translation System: ASURA. In *Eurospeech*, 1993.

Shiho Nobesawa et al. Segmenting Sentences into Linky Strings using D-bigram statistics. In *COLING*, Copenhagen, 1996.

Klaus Ries, Finn Dag Buø, and Alex Waibel Class Phrase Models for Language Modeling. In *ICSLP*, 1996.

Klaus Ries, Finn Dag Buø, and Ye-Yi Wang. Improved Language Modeling by Unsupervised Acquisition of Structure. In *ICASSP* 1995, Vol. 1, pp. 193-196.

Tanja Schultz and Detlef Koll. Spontaneously Spoken Japanese Speech Recognition with Janus-3 To appear in *EUROSPEECH*, 1997.

Peter Sells. VP in Japanese: Evidence from -te Complements. In *Japanese/Korean Linguistics*, pp. 319-333, SLA, Stanford, 1990.

Masayoshi Shibatani. Japanese. In *The World's Major Languages*, pp. 855-880, Bernard Comrie, ed., Oxford University Press, 1987.

Koichi Shinoda and Takao Watanabe. Speaker Adaptation with Autonomous Model Complexity Control by MDL Principle. In *ICASSP* 1996, Vol. 2, pp. 717-720.

Bernhard Suhm and Alex Waibel. Towards Better Language Models for Spontaneous Speech. In *ICSLP*, Yokohama, Japan, 1994.

B. V. Suhotin. Methode de dechiffrage, outil de recherche en linguistique. *TA Informationes*, 2:3–43, 1973.

Eiichiro Sumita and Hitoshi Iida. Heterogeneous Computing for Example-based Translation of Spoken Language. In *Proceedings of the sixth international conference on theoretical and methodological issues in Machine Translation*, Leuven, Belgium, 1995.

Virginia Teller and Eleanor Olds Batchelder. A Probabilistic Algorithm for Segmenting Non-Kanji Japanese Strings. In *AAAI* pp. 742-747, Seattle, 1994.

Kei Yoshimoto and Christine Nanz. A Study in Transfer Japanese-English. Verbmobil report 101 2/96.