A Language-Portable Orthography Analysis System

Barbara Wheatley, Thomas Staples, and Joseph Picone
Texas Instruments
Central Research Laboratories, P.O. Box 655474, MS 238, Dallas, Texas 75265, U.S.A.

The development and implementation of speech technology applications requires the ability to deal with both text and speech and to map between them. For example, text segmentation and mapping to pronunciation are required in order to automatically generate language models from text corpora for large vocabulary speech recognition; the reverse mapping, from pronunciation to text, is required to produce useful output from a large vocabulary recognition system. Text to pronunciation is also invaluable in speech corpus construction, both to develop phonetically balanced elicitation materials and to automatically time mark data that is transcribed orthographically.

A system that attempts to process multiple languages must deal with the fact that orthography is one of the most variable aspects of language. Languages differ in whether the basic orthographic unit corresponds to syntactic/semantic units, syllables, phonemes, or some combination or subset of these; in the regularity of the relationship between orthography and pronunciation; and in extent to which larger units such as words are indicated explicitly in text. Because of this variability, the nature of a problem such as text-to-pronunciation mapping may differ fundamentally from one language to another.

For example, a conventional approach for a language such as English is to use word-based dictionaries. Each dictionary entry is keyed by an orthographic word, defined in English as a space-delimited sequence of characters. And although the contents of entries may be highly complex, particularly in representing semantic and syntactic information, entries are accessed through a simple string matching procedure. However, such an approach may be much less suitable for a language like Japanese in which words are not delimited in normal text, or a language like Spanish in which text-to-pronunciation mapping is readily determinable without explicit enumeration of every word. In such cases, the conventional approach is to develop rule sets to supplement or replace dictionaries. For example, Japanese text may be preprocessed by a rule-based parser that attempts to determine word boundaries. Spanish text may be processed using a rule set (such as the text-to-pronunciation component of a text-to-speech system) to generate pronunciations.

The major drawback of this strategy is the difficulty of supporting it readily in a new language. Rule-based approaches may provide the most parsimonious and elegant solutions, but they also typically require a high level of expertise to develop and maintain. It can be quite difficult to find individuals who combine this kind of expertise with knowledge of the language. Also, implementing rule-based systems in a generic way that supports ready extension to other languages is a major challenge, requiring substantial knowledge and effort in system design.

As an alternative to rule-based systems, we have developed a method that combines dictionaries with a dynamic programming algorithm to support diverse languages in a uniform system. In this system, dictionaries are not necessarily word-based; instead, entry keys are simply defined to consist of sequences of symbols (n-grams). When the dictionary is used to process any given input, for example to map a text sentence to pronunciation, the dynamic programming algorithm performs a length-constrained n-gram search to find the optimal set of entries for that string, as shown in Figure 1. Each n-gram in the dictionary has an associated weight that is used in determining which matches are optimal. This weight is normally defined to be a simple function of the entry length, such that the search algorithm will favor longer matches over shorter ones.

The n-gram dictionary system has been used for Japanese to perform both segmentation of sentences into words and mapping from orthography to pronunciation. Dictionary entries consist of n-grams of logographic characters (kanji) and syllabary symbols (kana). Entries are chosen to provide enough information to reliably perform segmentation and pronunciation mapping. The pronunciation of kanji is typically context-sensitive and may depend on word segmentation. Entries may span word boundaries in order to provide the necessary context, so that a single entry may contain multiple words or even fractional words—such as a kana particle or suffix followed by a word-initial kanji. Also, entries may be partial words, facilitating efficient treatment of verb stems and conjugations. In tests of accuracy in converting text sentences to pronunciation, this system performed significantly better than public domain tools, as shown in Table 1.

To support word segmentation, each entry contains word boundary information. Because the correct segmentation for an entry may depend on the larger context in which it occurs, the boundary labels distinguish between definite word boundaries and potential word boundaries. Uncertainties are resolved by a simple postprocessor which looks at adjacent boundary labels arising from concatenation of dictionary entries to determine which potential boundaries are realized. For example, an entry which might be a complete word or might appear with a suffix is marked as terminating in a potential word boundary; the suffix entry has no initial boundary, either potential or definite. When these occur together, the potential boundary is not realized and the entire unit is classified as one word. Thus, word segmentation of text is performed using the dictionary and n-gram matching algorithm; no rule-based parsing is necessary.

The n-gram system has also been used to process Spanish text. In this case, the problem is vastly simpler: given a small amount of context, letter-to-sound mapping in Spanish is predictable, with a small number of exceptions such as place names. Both exceptional and regular mappings in Spanish are easily handled by the n-gram dictionary. Each entry contains one or more symbols consisting of letters or word boundaries (which are marked in Spanish orthography); weights are assigned to favor longer sequences and exceptions. The search algorithm finds the

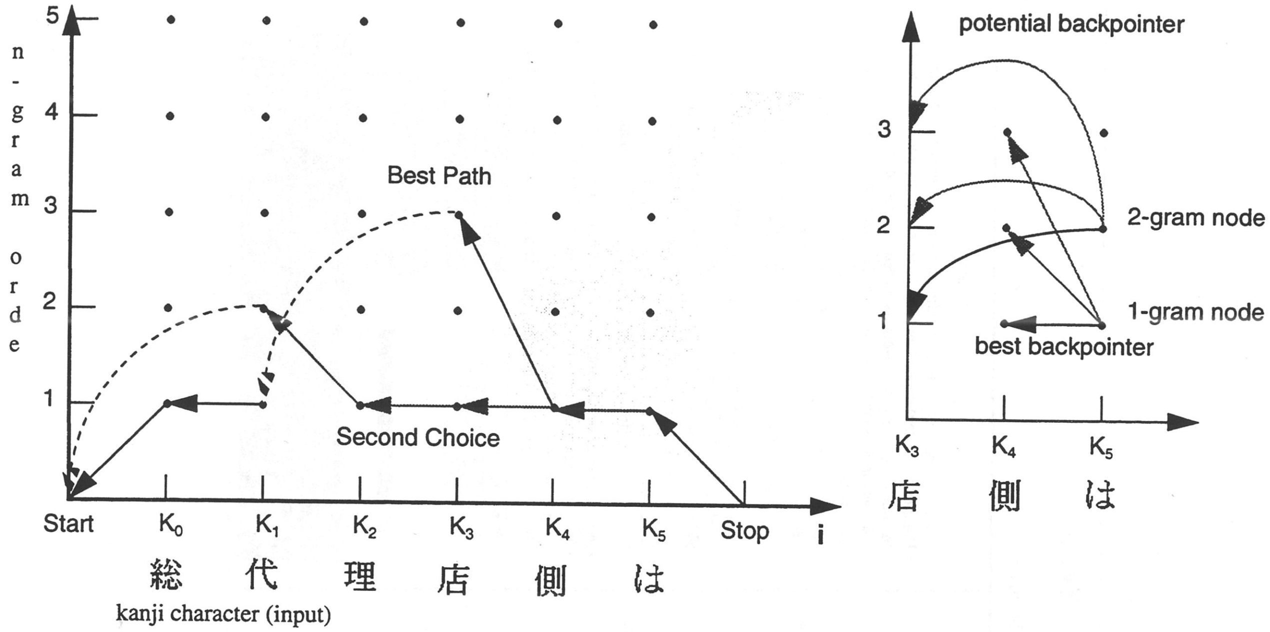


Figure 1. An example illustrating an efficient dynamic programming-based search for the best dictionary match. The horizontal axis corresponds to the input characters. The vertical axis corresponds to the *n*-gram order. Node costs include the weight of an appropriate entry in the dictionary. Straight lines denote backpointers that point to the previous column; arcs denote backpointers that skip the previous column(s). Previous paths from a given node can extend backwards as far back as the maximum *n*-gram order. Theoretically, this would be prohibitively expensive in computational cost. In practice, there are very few competing paths that need to be considered. A diagram showing the first three allowable backpointers for a 1-gram and 2-gram node is shown to the right.

optimal set of entries for any given input word or sentence and produces the expected pronunciation.

This system has been used to process Spanish text corpora to produce phonetically balanced sentence sets. It also lends itself to other uses. It can serve to bootstrap a word-based dictionary from text, providing possible orthographies corresponding to phoneme sequences, to support large vocabulary recognition in Spanish. Also, the same system (with a different dictionary) has been used to divide words into units corresponding to named letters in Spanish—which includes **ch** and **ll**, as well as individual characters. This facilitates construction of elicitation materials to collect a corpus of spelled words in Spanish.

The n-gram dictionary system is used to provide real-time support for an interactive transcription and time marking tool for multilanguage speech corpus creation. The tool provides a multilanguage editor, waveform display and audio functions, integrated with automatic time marking. Time marking is performed using supervised recognition, with supervision based on the transcription. Integration of the n-gram dictionary system allows transcription in native orthographies, since the orthography can be readily converted to the expected pronunciation.

In summary, the n-gram dictionary system allows a uniform approach to be used for text-to-pronunciation mapping in languages as divergent as Japanese and Spanish. More importantly, the approach does not rely on a high degree of expertise in syntactic parsing or complex text-to-pronunciation rules. Creating and maintaining the dictionaries requires expert knowledge of the language, but only the kind of knowledge typical of an educated, intelligent native speaker—not a native speaker who is also a speech technologist or parsing expert.

Algorithm	Combined Sentence Error Rate	Sentence Substitution Error Rate	Sentence Rejection Error Rate	Kanji Character Substitution Error Rate
JUMAN	39.2%	30.3%	8.9%	9.0%
Wnn	52.2%	20.1%	32.1%	21.8%
KAKASI	11.1%	11.0%	0.1%	1.8%
N-Gram	3.6%	3.6%	0.0%	0.5%

Figure 1. A summary of the results of evaluations on a 1,000 sentence database. The *n*-gram algorithm, in closed-set testing, is shown to provide significantly better performance than public domain counterparts.

A Language-Portable Orthography Analysis System

Barbara Wheatley, Thomas Staples, Joseph Picone
Texas Instruments, Systems and Information Science Laboratory, Dallas, Texas, USA

Objective

- Text/speech mapping for speech recognition development
- Coherent system to support diverse languages
- Maintainable/extensible by educated native speaker
- Minimal pre- and post-processing requirements
  =>Limit parsing, rule design and management
  =>Trade computational resources for human expertise

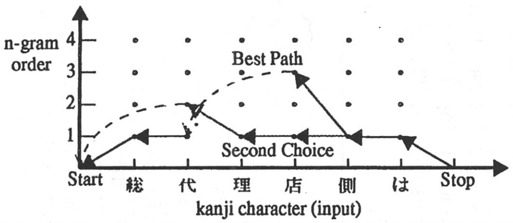
Approach

- Dictionaries to enumerate context-sensitive mappings
- Units are arbitrary n-grams, chosen to preserve necessary context (cross-word, word, sub-word as needed)
- Analysis of given text uses dynamic programming algorithm to find best match via exhaustive search

Japanese text example: 総代理店側は

Left-to-right/largest-word-first:				
総代	理	店	側	は
そうだい	り	みせ	がわ	は

Exhaustive search (correct):			
総	代理店	側	は
そう	だいりてん	がわ	は



Japanese

- 3 writing systems: 1 logographic (kanji) and 2 syllabaries
- Pronunciation of kanji is highly context-dependent
- Words are not indicated in text

Pronunciation and Segmentation:

- N-grams provide sufficient context to disambiguate pronunciation
- Symbols in entries mark potential (&) or actual (#) word boundary positions
- Examples:

#移行#	#いこう#	#ik(ou|o:)#	2.01
#行く&	#いく&	#iku&	2.01
&行&	&ぎょう&	&gyo:&	1.00
#歩き	#あるき	#aruki	2.01
し#人#	し#ひと#	shi#hito#	2.01
&人&	&じん&	&jing&	1.00

- Simple post-processor resolves symbols concatenated by dictionary search:

```
(Input)  救急車が十分に動けず救助作業が遅れている
(Output) # kyu: kyu: sha # @ ga @ # ju: bu ng # @ ni @ # u go
               ke & & zu & # kyu: jo # # sa gyo u # @ ga @ # o
               ku re & & te & # i ru #  (20.080000)
kyu:kyu:sha ga ju:bung ni ugokezu kyu:jo sagyou ga okurete iru
```

Coverage:

- 145,000 entries, from 1-grams to 9-grams
- Comparison with public-domain systems: accuracy in closed-set test on FJ news (1000 sentences from electronic news groups)

Algorithm	Sen Err	Sen Subs	Sen Rej	Char Subs
JUMAN	39.2%	30.3%	8.9%	9.0%
Wnn	52.2%	20.1%	32.1%	21.8%
KAKASI	11.1%	11.0%	0.1%	1.8%
N-Gram	3.6%	3.6%	0.0%	0.5%

Spanish

- Nominal pronunciations are predictable from letter combinations
- Enumerate context dependent mappings (generally sub-word units)
- Include exceptions, with weights biased to select them
- Use system to bootstrap word-based dictionary from text, if desired
- Examples:

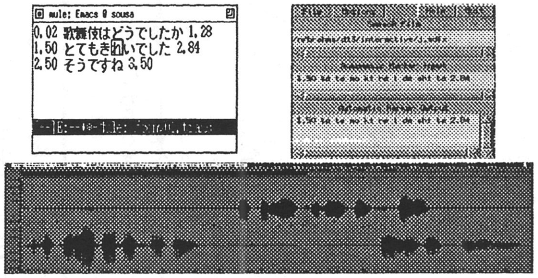
r	r(	1.00	x	s	1.00
#r	r	2.01	xa	ksa	2.01
rr	r	2.01	México	mexiko	6.05

=>Same dictionary format and algorithm provide effective support for extreme cases: Japanese and Spanish

=>Unifying element: context sensitive n-grams, not necessarily word-based, with efficient pattern-matching look-up algorithm

Multilanguage Transcription/Alignment

- Application of orthography analysis system: "Timesaver" tool
- Supports transcription in native orthography with real-time automatic alignment based on pronunciation derived from orthography
- Tool will be used to transcribe and time align "Call Home" corpus



```
0.02 歌舞伎はどうでしたか 1.28
1.50 とてもきれいでした 2.84
2.50 そうですね 3.50
```