A Comparative Study of Unsupervised Grapheme-Phoneme Alignment Methods

Timothy Baldwin and Hozumi Tanaka

Dept of Computer Science

Tokyo Institute of Technology

2-12-1 Ookayama,

Meguro-ku, Tokyo 152-8552 JAPAN

{tim,tanaka}@cl.cs.titech.ac.jp

Abstract

This paper describes and compares two unsupervised algorithms to automatically align Japanese grapheme and phoneme strings, identifying segment-level correspondences between them. The first algorithm is inspired by the TF-IDF model, including enhancements to handle phonological variation and determine frequency through analysis of "alignment potential". The second algorithm relies on the C4.5 classification system, and makes multiple passes over the alignment data until consistency of output is achieved. In evaluation, the first algorithm proves to be greatly superior to the second, producing a word accuracy of 96.94%.

Introduction

The task of grapheme-phoneme alignment is intrinsically related to text-to-speech conversion, and provides the basic toolset of grapheme-phoneme correspondences for use in predicting the pronunciation of a given word. While it is certainly possible to handcraft grapheme-to-phoneme mappings (see, e.g., (Allen et al., 1987; Sejnowski and Rosenberg, 1987; Huang et al., 1994; Divay and Vitale, 1997)), we suggest that it should be possible to automatically extract such data from a database of grapheme-phoneme string pairs *without any form of supervision*. Thus, given a pronunciation-annotated machine-readable dictionary, it should be possible to generate a set of aligned grapheme–phoneme (word–pronunciation) pairs reliably and fully automatically. Theoretically, the grapheme-phoneme alignment output could then be plugged into a reading machine, producing an instant text-to-speech system for any language (as per (Ling and Zhang, 1998; Black et al., 1998)).

The objective of this paper is to analyse the applicability of unsupervised learning methods to automated grapheme-phoneme alignment in Japanese. In particular, we propose an incremental learning algorithm founded upon the TF-IDF metric, and compare this to a multi-pass alignment method drawing on the C4.5 classification system (inspired by the method of (Ling and Wang, 1997)). Alignment data is first constructed by exhaustively generating all alignment mappings for a given grapheme-phoneme pair. We filter off lexically and phonologically implausible alignment candidates from this data, and feed the final set of alignment candidates into the different alignment algorithms. These algorithms then incrementally disambiguate the data to produce a unique alignment candidate for each grapheme-phoneme tuple, through analysis of frequency distribution in the data.

Definitions

Japanese is made up of the three native orthographies of kanji, katakana and hiragana. **Kanji** characters (e.g.

"消") derive from the Chinese writing system and are largely ideographic in nature; a single kanji character tends to have multiple pronunciations (a sample of readings for "消" include $sy\bar{o}$, $ki(eru)$ and $ke(su)$). **Katakana** and **hiragana** (collectively described as **kana**) are isomorphic syllabaries, with each character describing a unique, mutually exclusive phoneme content; examples of hiragana and katakana are "し" (si) and "ゴ" (go), respectively. The three orthographies intermingle in modern-day Japanese texts, with hiragana generally used for inflectional affixes, case particles and stop words, katakana for loan words, and kanji for content word stems. This effect is seen in the word 消しゴム [$kesigomu$] "eraser", which incorporates all three script types.

In targeting "graphemic Japanese", therefore, we must consider all three writing systems. Phonemic Japanese, on the other hand, can be described through kana characters, as all kanji characters are transcribable into kana, and kana describe the full phonemic inventory of Japanese in the form of phoneme chunks. That is not to say that every kana character maps to a single phoneme, but there is a unique broad phonetic transcription associated with almost all kana characters.[1] It is thus trivial to complete the full grapheme-phoneme conversion process if necessary, and at the same time, our choice of kana characters as phoneme medium frees us from consideration of low-level connectional restrictions between phoneme units, as this information is implicitly encoded within the orthography.

Grapheme-phoneme ("G-P") alignment is defined as the task of *maximally* segmenting a grapheme compound (a single dictionary entry, usually constituting a single word) into morpho-phonic units, and aligning each such unit to its corresponding phoneme unit in the phonetic transcription for that compound. Segmentation of the grapheme compound is maximal in the sense that no segment can be further segmented into aligning subsegments. To take the example of the grapheme string 感-謝-su-ru [ka-n-sya-su-ru] "to thank/be thankful",[2] 感 aligns with ka-n in the phoneme string, and 謝 with sya, as indicated in $align_1$ of Figure 1.

---

[1] The only exception to affect us is the kana u, which when not used as an inflecting suffix, is pronounced as /o/ when immediately proceeding an /o/ sound within a phoneme segment, and /u/ otherwise. Here, disambiguation is possible given phoneme segment context and part-of-speech information.

[2] So as to make this paper as accessible as possible to readers not familiar with Japanese, kana characters are written italicised in Latin script for the remainder of this paper, with character boundaries indicated by "-" and segment boundaries (which double as character boundaries) indicated by "⊙".
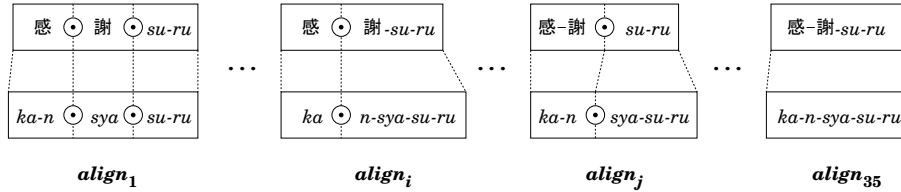
Figure 1: Candidate alignments for 感-謝-su-ru [*ka-n-sya-su-ru*] "to thank/be thankful"

Cognitive aspects of G-P alignment

One vital issue in grapheme-phoneme alignment is the determination of 'atomic' grapheme segments, that is segments which are not further divisible. Clearly, the lower bound on atom size for Japanese is a single kana or kanji character, but there is no inherent upper bound on the number of characters that can combine to form a segment, for either grapheme or phoneme segments. While it is correct to say that there is a cognitive preference to segment off individual kanji characters (possibly with kana suffices), there is equally potential for (indivisible) multiple-kanji grapheme segments, such as 台-詞 [*se-ri-fu*] "one's lines". Consequently, alignment does not simply consist of segmenting the grapheme string up into individual characters and aligning them with chunks of the phoneme string, and consideration must be given to the granularity of segmentation.

A number of inter-related cognitive factors seem to determine the "segmentability" of a grapheme string and resultant "alignability" with a given phoneme string, namely: (i) the relative frequency of each segment-level G-P sub-alignment; (ii) the cognitive immediacy of adjacent segments; and (iii) phonetic similarity to regular readings in the case of novel G-P sub-alignment.

High relative frequency of alignment refers to the situation of a given grapheme segment g commonly aligning with a given phoneme segment p (and phonological variants thereof), such as 感 invariably aligning with the reading *ka-n*. Clearly if the $\langle...\odot g\odot...\rangle - \langle...\odot p\odot...\rangle$ alignment sub-schema is observed with sufficient frequency, a natural preference will arise to emulate that same alignment sub-schema wherever possible, for reasons of familiarity.

In the case that there is no alignment schema which produces familiar alignments for all individual grapheme segments, there is a tendency to preserve as much regularity to the overall alignment schema as possible by maximising the number of regular alignments and framing any irregular alignments between segment-level alignments of high cognitive immediacy. Thus, when presented with a G-P tuple such as $\langle$ 白髪$\rangle - \langle si\text{-}ra\text{-}ga\rangle$, where 白 is commonly associated with the reading *si-ra* but not *si* and there are no independent instances of 髪 taking a *ga* or *ra-ga* reading, there is a natural preference to uphold the single known sub-alignment for 白 and produce a forced alignment for 髪, as in $\langle$ 白$\odot$髪$\rangle - \langle si\text{-}ra\odot ga\rangle$.

Finally, if a novel alignment must be made such as $\langle...\odot$髪$\rangle - \langle...\odot ga\rangle$ above, conservatism rules in that irregular readings tend to be chosen so as to be phonetically similar to established readings. In the case of 髪, the established reading is *ka-mi* (or *ga-mi* in its voiced realisation), from which the deletion of a single character produces the suggested *ga* reading.

In the case that the above processes do not apply to any substring of the G-P tuple, the tendency is to chunk unalignable kanji together into a single multi-kanji segment, such as occurred for *se-ri-fu* above.

The implications of the above observations to our statistical modelling of G-P alignment are to develop a model which gives preference to sub-alignments of high plausibility, allows irregular alignments given that the surrounding context displays high cognitive immediacy of alignment, and has the facility to "back-off" to multi-kanji segments when necessary. Our interpretation of TF-IDF is suggested to constitute such a model.

Grapheme-phoneme alignment

Grapheme-phoneme alignment is performed as a three-stage process: (a) detection of lexical alternation and removal of lexical alternates from the input; (b) determination of all possible alignment candidates and subsequent pruning through alignment constraints; and (c) scoring of all final candidate alignments to determine the final solution.

Lexical alternates are defined as containing the same kanji characters in the same linear order, and coinciding in phonemic content (i.e. having the same reading). We enforce the constraint that all lexical alternates must be governed by the same basic alternation schema, allowing us to filter off alignment candidates for a given G-P tuple which are incompatible with one or more alternates of that tuple.

Given that both grapheme and phoneme segments can be of arbitrary length, **alignment candidate generation** must encompass all segmentation cardinalities. That is, for the example of a three-character grapheme string, we must consider the maximal segmentation of the string into three segments, and also partial segmentations into two segments or alternatively a single segment encompassing the full string.

Luckily, we are able to rely on strict linearity of alignment between the grapheme and phoneme strings, and in most cases can count on the alignment being isomorphic (the only exception being "grapheme gapping" – see (Baldwin and Tanaka, 1999b)). As a result, the total number of alignments is given by $\sum_{x=0}^{l-1} C_x^{m-1} C_x^{n-1}$ in the general case, where m is the character length of the grapheme string, n the character length of the phoneme string, and $l = \min(m, n)$.

We are able to reduce the alignment space considerably, however, through the advent of five lexical and phonological constraints on alignment, as described in (Baldwin and Tanaka, 1999b). These constraints apply to script and syllable boundaries, the character length of aligned segments, and the number of voiced obstruents contained in a single phoneme segment. For the data targeted in evaluation, the average alignment paradigm size was reduced from 12.06 to 3.27 (a reduction of close to 75%), with no instances of the correct alignment candidate being pruned from the alignment paradigm.

It is important to realise that the application of the above constraints not only reduces the search space for statistical scoring, but can actually single out a unique

$$freq(\langle g,p\rangle) = \left| \left\{ \langle GS, PS\rangle : \exists p' \in phon\_var(p)\left\{ \langle ...\bigodot_i g\bigodot_{i+1}...\rangle - \langle ...\bigodot_i p'\bigodot_{i+1}...\rangle \in \{\langle GS_{seg}\rangle - \langle PS_{seg}\rangle\} \right\} \right\} \right| \quad (1)$$

$$wfreq(\langle g,p\rangle) = SOLVED \cdot freq_{SOLVED}(\langle g,p\rangle) + UNSOLVED \cdot freq_{UNSOLVED}(\langle g,p\rangle) \quad (2)$$

$$tf\text{-}idf(\langle g,p,ctxt\rangle) = \underbrace{\frac{wfreq(\langle g,p\rangle) - UNSOLVED + \alpha}{wfreq(\langle g\rangle)}}_{tf(\langle g,p\rangle)} \underbrace{\log\left(\frac{wfreq(\langle g,p\rangle)}{wfreq(\langle g,p,ctxt\rangle) - UNSOLVED + \alpha}\right)}_{idf(\langle g,p,ctxt\rangle)} \quad (3)$$

legal solution, providing what turns out to be vital "free ride" alignment data to bootstrap the different systems with.

The alignment constraints are the only component of the overall formulation which is specific to Japanese, and the different algorithms would be equally applicable to unconstrained alignment data, making them directly transferrable to any other language.

We next turn to description of the two main unsupervised alignment methods.

Incremental learning with TF-IDF

The first algorithm (originally proposed in (Baldwin and Tanaka, 1999a)) is based on incremental learning or "hill-climbing", whereby the system disambiguates a single alignment paradigm at a time and incrementally updates the statistical model according to both discarded alignment candidates and the selected alignment solution. Selection of the alignment paradigm to be disambiguated is performed according to an adaptation of the TF-IDF scoring metric (Salton and Buckley, 1990), originally developed within the information retrieval fraternity for term weighting. In this, we score and rank each alignment candidate contained within the current alignment paradigm, and further rank the different alignment paradigms according to the weighted ratio between the top- and second-ranking alignment candidates. The highest-scoring alignment paradigm on each iteration is selected for disambiguation, according to the top-ranking alignment candidate described therein. We then update the statistical model, revise scores for alignment paradigms affected by the changed statistics, and rerank in preparation for the next iteration.

The utility of TF-IDF within the task of G-P alignment, stems from it weighting up terms (aligned G-P segments) which occur frequently within a given document (grapheme segment) context, but relatively infrequently within other documents (left/right adjoining grapheme and phoneme contexts). As described above, we wish to model the cognitive process of G-P alignment by maximally weighting high-frequency (regular) readings for a given grapheme string, but at the same time scoring down readings which occur primarily in a fixed lexical context, as this would tend to point to oversegmentation at the phoneme level (the phoneme context is in actual fact part of the reading for the current grapheme segment) and/or the grapheme level (the grapheme context clusters with the current grapheme segment to form a multiple-grapheme segment).

In addition to facilitating the detection of regular alignments, TF-IDF provides a means of variably "windowing" over the grapheme and phoneme strings, in that it does not involve a pre-conceived notion of segment size. Additionally, by way of taking the mean of the scores for the left/right and grapheme/phoneme contexts for each aligned G-P segment pair, we are able to weight up alignments with more highly regularised segment-level

readings, again mirroring the cognitive processing of G-P alignment.

While TF-IDF offers no immediate solution to the third cognitive issue of conservatism in cases of non-regular readings, it does allow us to handle abbreviations of regular readings—as was seen above for the *ga* reading of 髪—in that they will generally be found within the (undisambiguated) alignment paradigm of G-P tuples drawing on the regular reading.

Counting frequencies

Clearly, to be able to apply the TF-IDF metric, we require some way of counting frequencies. For disambiguated alignment paradigms, we can rely on the absolute frequencies of segments contained within the alignment solution. For residue alignment paradigms awaiting disambiguation, on the other hand, we have an arbitrary number of alignment candidates to choose from, and no immediate way of producing an all-encompassing frequency value.

We resolve this issue by associating a single frequency count with every segment type occurring independently in a given alignment paradigm, irrespective of the number of alignment candidates it occurs within. In this way, we model the "alignment potential" of each segment. This process can be formalised as in equation equation (1), in the case of $freq(\langle g,p\rangle)$ (singleton and triple segment combinations are defined in a similar fashion). Here, p is the phoneme segment aligning with grapheme segment g, and $phon\_var(p)$ describes the set of "phonological alternates" of p. Phonological alternates are predictable instances of phonological alternation from a base form p, with the most widespread types of phonological alternation being "sequential voicing" (Tsujimura, 1996) and gemination. Fortunately, phonological alternation occurs only on syllables at phoneme segment boundaries, and phonological alternate "equivalence classes" are mutually exclusive in the main. This allows us to cluster together frequencies for members of each equivalence class, going some way toward combating the effects of data sparseness.

The basic TF-IDF model

Our interpretation of the TF-IDF model is given in equation equation (3), where g is a grapheme segment, p a phoneme segment and *ctxt* a single phoneme or grapheme context for $\langle g,p\rangle$ within the current alignment. As an additional facet of hill-climbing, we weight up segment frequencies contained within disambiguated alignment paradigms ($freq_{SOLVED}$), over those for unprocessed alignment processes ($freq_{UNSOLVED}$). This is achieved through the *wfreq* functions for the various segment combinations, which use the fixed *SOLVED* and *UNSOLVED* weights to prioritise disambiguated frequencies ($0 < \alpha < UNSOLVED \leq SOLVED$). The subtractions by a factor of *UNSOLVED* are designed to discount the single occurrences of $\langle g,p\rangle$ and $\langle g,p,ctxt\rangle$ in the current align-

ment paradigm, and α is an additive smoothing constant designed to counter the effects of low frequency counts.

As described above, consideration of lexical context for a given segment tuple $\langle g, p \rangle$ is four-fold, made up of the single *character* immediately adjacent to g in the grapheme string and single *syllable* immediately adjacent to p in the phoneme string (or the null string in the case of a string-initial/final segment), for both the left and right directions. An individual *tf-idf* score is calculated for each of these contexts *ctxt*, and the resultant scores combined by taking the 4-way arithmetic mean. In the case of full-string segment alignment, the overall score is defined to be $tf(\langle g, p \rangle)$.

The overall score for all G-P segment tuples contained in the current alignment is computed according to the arithmetic mean of the respective combined *tf-idf* scores, with the proviso that full kana-based grapheme segments are excluded from computation.

Additional allowances for affixation and conjugation are made according to the method described in (Baldwin and Tanaka, 1999b).

Selective sampling

As described above, a single alignment paradigm is selected for disambiguation on each iteration, and the statistical model updated by way of incrementing frequencies for segment alignments contained in the alignment solution (according to the *SOLVED* weight), and decrementing frequencies deriving from segments occurring in disallowed alignment candidates and not the alignment solution. The method of "selectively sampling" a single alignment solution on each iteration, is performed by calculating a score for each alignment paradigm, and disambiguating the highest scoring paradigm according to the top-scoring alignment candidate contained therein. The score for the alignment paradigm is calculated according to the "weighted log odds" discriminative ratio $s_1 \log \frac{s_1}{s_2}$, where s_1 is the score for the top-ranking alignment candidate and s_2 that for the second-ranking alignment candidate within the current alignment paradigm. Intuitively, this balances up maximisation of both s_1 and the disparity between s_1 and s_2, such that we are after not only alignment candidates which score well, but also alignment paradigms where the top-scoring alignment candidate has a clear empirical advantage over other candidates.

Multi-pass classification

The second algorithm is inspired by the research of Ling and Wang (1997), who applied the C4.5 classification system (Quinlan, 1993) to unsupervised alignment of English G-P tuples. Specifically, C4.5 was used to predict the phonemic equivalent of English words (graphemic strings), by way of outputting a phoneme for each constituent character in a given character window and combining phonemes to give an aligned phonemic equivalent for the original word. A phonetic transcription for the original word was then used to independently generate alignment candidates, and the alignment candidate most similar to the C4.5-constructed alignment chosen as the alignment solution. Similarly to our incremental learning method, alignment solutions are then fed back into C4.5 as training data, for use in aligning subsequent words. Ling and Wang implemented a number of heuristics to improve the performance of their basic method, including ordering the system inputs in ascending order of alignment cardinality, delaying making a decision in cases of multiple alignment candidates being equally similar

to the C4.5-constructed alignment, and cross-validating held-out partitions of the alignment data against the remainder of the data. The final alignment precision over 33,121 English words exceeded 99.5%.

The alignment accuracy on English is certainly impressive, and suggests the method as promising for Japanese G-P alignment. Unfortunately, however, the case of Japanese G-P alignment is considerably more complex than that of English. Most importantly, as noted above, Japanese phoneme segments often extend over multiple characters for single character grapheme segments even, whereas in the case of English, grapheme segments almost always map onto single phonemes. It was thus possible for Ling and Wang to enumerate the 40 or so possible phoneme segments and have C4.5 choose between them in predicting the phonemic equivalent of each grapheme segment. If we attempted to do the same for Japanese, we would end up with a total of over 56,000 phoneme segments in the case of the data set used in evaluation, making the classification task unmanageable. Additionally, English uses only 26 letters (assuming uniform case), whereas our test data contains 4429 grapheme character tokens and 167 phoneme character tokens. This blow out in the search space suggests the need for a different classification approach.

On the implementation side, Ling and Wang are unable to use negative evidence from discarded alignment candidates, a possibility we look to. We also use the certainty factor values returned by C4.5 in scoring the plausibility of different alignment candidates.

Algorithm basics

We apply the basic fixed window method suggested by Ling and Wang, but instead of inputting only grapheme context to return a phoneme, we input both grapheme and phoneme contexts to return a binary judgement on the plausibility of a coincident segment boundary existing at the centre of the two context windows. Grapheme and phoneme context is set to 3 characters on either side of the segment boundary, making for a combined window size of 12 characters. To give an example based on $align_1$ from Figure 1, the classifier "_, _, 感, 謝, *su, ru*, _, *ka, n, sya, su, ru*" should produce a judgement of true, corresponding to the leftmost inter-segment boundary in $align_1$ (the '_' token indicates an empty character beyond the boundaries of the original word, and the underlined component is the grapheme window). "_, _, 感, 謝, *su, ru, ka, n, sya, su, ru*, _", on the other hand, is associated with a negative judgement within the context of $align_1$, as despite segment boundaries existing at the centres of the two context windows, they do not coincide under alignment.

At the same time as classifying input for segment boundary compatibility, C4.5 can be set to output a certainty factor $cf : 0 \leq cf \leq 1$ for each class. This is particularly useful in comparing the overall plausibility of alignment candidates with little or no segment boundary overlap. In the case of the tuple 大使 [*ta-i-si*] "ambassador", for example, we need to choose between the three alignment candidates of $a_1 = \langle$ 大⊙使$\rangle$-$\langle ta$⊙i-$si \rangle$, $a_2 = \langle$ 大⊙使$\rangle$-$\langle ta$-i⊙$si \rangle$ and $a_3 = \langle$ 大-使$\rangle$-$\langle ta$-i-$si \rangle$, the second of which (a_2) is correct. In sizing up a_3 against a_1 and a_2, we are making a judgement as to the plausibility of the single segment boundary distinguishing each alignment candidate pairing. However, if a_1 were determined to be more plausible that a_3, and equivalently a_2 more plausible than a_3, how could we choose between a_1 and a_2? Here, we apply the certainty factors in transferring evaluation across to a numeric comparison.

So as to limit comparison of alignment candidates to only those determined to be legal by the alignment constraints, we cluster "homogeneous" legal alignment candidates together into "packed alignment arrays" and individually score each alignment candidate described therein. **Packed alignment arrays** are of the form ⟨ 大?使⟩–⟨*ta?i-si*⟩, for example, where '?' indicates an optional segment boundary aligning with the corresponding boundary in the opposing string (note that packed alignment arrays can also contain fixed segment boundaries, the score for which contributes to the overall score of alignment). **Homogeneous** alignment candidates are defined as not having any crossing-over of alignment and having all coincident segments aligning similarly. a_1 and a_2 from above are not homogeneous (due to the "大" and "使" segments aligning differently), producing the two packed alignment arrays ⟨ 大?使⟩–⟨*ta?i-si*⟩ and ⟨ 大?使⟩–⟨*ta-i?si*⟩ for "大使". A combined score for each alignment candidate is determined based on the average segment boundary certainty factor, and alignment candidates realised in multiple packed alignment arrays (such as a_3 in the two presented arrays) are given the minimum score out of those realisations. The optimal alignment candidate for a given alignment paradigm is then that which produces the maximum mean certainty factor.

While we do not have any pre-annotated training data (preserving the true unsupervised status of our method), we do have disambiguated positive evidence from the "free ride" data disambiguated by the alignment constraints. We can also construct negative evidence from alignment candidates disallowed by the alignment constraints,[3] although here there is potential for segment-level overlap with the correct alignment. We thus take only those segment boundary windows not found within the final alignment paradigm. Finally, we stretch the boundaries of unsupervised learning somewhat in providing the system with positive instances for each hiragana and katakana character, aligning it as a single character in the grapheme and phoneme strings. These instances combine to form the "bootstrap data" with which the system is initialised.

We further order the G-P tuple data set in ascending order of alignment cardinality, in the manner of Ling and Wang, so as to give the system the chance to make easier decisions early on and use the resulting evidence in making decisions of greater complexity later on.

First pass
In the first pass over the data, C4.5 is initialised with the bootstrap data from above, and run over each alignment paradigm in turn. The classifying decision tree is then updated each time an alignment paradigm is disambiguated, based on the positive evidence described by the alignment solution. Due to the potentially dubious nature of evidence arising from this first pass, we commit ourselves to an alignment solution only in the case that there is no tie in combined certainty factor; in the case of a tie, we reserve our decision for subsequent passes. Additionally, we feed only positive evidence back into C4.5.

Second and subsequent passes
In the second and subsequent passes, we classify each alignment paradigm according to the combination of the

---

[3]We take only negative evidence produced through alignment incompatibility between otherwise legal alignment candidates for lexical alternates, and also that produced by Lyman's Law (Vance, 1987).

original bootstrap data and any positive or negative data generated for other alignment paradigms in the preceding pass, holding out data pertaining to the current alignment paradigm. Negative data is progressively added into the training data throughout the second pass, and maintained through subsequent passes.

In the case of a tie in alignment score, we take the first alignment producing that score. Note that packed alignment arrays are listed in descending order of the number of individual alignment candidates they describe, such that by taking the first alignment candidate to produce the maximum alignment score, we are giving it credit for having won out over a larger number of alignment candidates.

We continue iterating over the data until the combined alignment output converges, that is we attain consistent output over successive passes.

Evaluation
The proposed systems were tested on a set of 5000 G-P tuples containing at least one kanji, randomly extracted from the combined EDICT Japanese-English[4] and Shinmeikai (Nagasawa, 1981) dictionaries. Any lexical alternates of the 5000 G-P tuples were further added to the test set to give the alignment constraints full scope for application (expanding the test data out to 6503 instances), and the original 5000 G-P tuples manually aligned for use in system performance evaluation. The extra lexical alternate data is used only in applying the alignment constraints and has no bearing on subsequent evaluation. The annotated alignment data was, of course, not available to any of the system configurations at execution time.

The test data was additionally pre-processed into alignment paradigms and sorted into ascending order of alignment cardinality, so as to ensure that the input to the two systems was identical.

The baseline word accuracy for this test suite, based on random selection of an alignment schema from the final alignment paradigm for each G-P tuple, is 44.75%.

Looking first to the incremental TF-IDF learning method, we tested the algorithm with different settings for the parameters *SOLVED*, *UNSOLVED* and α, and found that the respective values of 1.0, 0.5 and 0.05 produced the best word accuracy of 96.94%. Higher values of α tended to produce greater levels of under-alignment (chunking together of grapheme and phoneme segments into single super-segments), whereas lower levels of α produced greater levels of over-alignment (intra-segment segmentation). Larger values of *SOLVED*, on the other hand, tended to inflate the overall rate of both over- and under-alignment. Interestingly, most errors were homogeneous with the correct alignment.

The multi-pass classification method produced a word accuracy of 47.34% on the first pass and 58.18% on the second and third passes, halting on completion of the third pass due to coincidence of output with the second pass. Ties in alignment score were produced for only 78 of the 5000 annotated G-P tuples on the first pass, such that we were able to produce alignments for 4922 G-P tuples. On the first pass, most errors took the form of underalignment, whereas errors on the second and third passes were generally instances of overalignment or non-homogeneous with the correct alignment.

Based on these figures, the incremental TF-IDF learning method is clearly superior to the multi-pass classification

---

[4]ftp://ftp.cc.monash.edu.au/pub/nihongo

method, both in terms of raw accuracy and the degree of error in the case of incorrect output. The 58.18% word accuracy for the multi-pass classification approach also contrasts starkly with the 99.5% word accuracy claimed by Ling and Wang for English G-P alignment, although it does well outperform the baseline word accuracy.

To further compare the accuracies of the different methods, we calculated the progressive alignment accuracy over corridors of 250 alignment solution outputs, based on the order of output. In the case of the multi-pass classification method, the order of output corresponds to the order of the original data, in increasing order of alignment cardinality, whereas for the incremental TF-IDF learning method, the order of output is determined by the discriminative ratio values. For both methods, however, the first 895 outputs are the "free ride" alignment paradigms of cardinality one, at word accuracy 100%. The progressive word accuracies for pass 1 and passes 2 and 3 of the multi-pass classification method are presented separately as "MP-P1" and "MP-P2/3", respectively. We include evaluation of a number of variations on the basic TF-IDF method ("BASIC") to verify the efficacy of the discriminative ratio, firstly by way of a random sampling method, where a random alignment paradigm is disambiguated at each iteration ("RAND"), and secondly by way of a non-incremental method where all alignment paradigms are disambiguated according to the initial top-scoring alignment candidate and output in descending order of the discriminative ratio value ("DUMP"). The various progressive word accuracies are given in Figure 2.
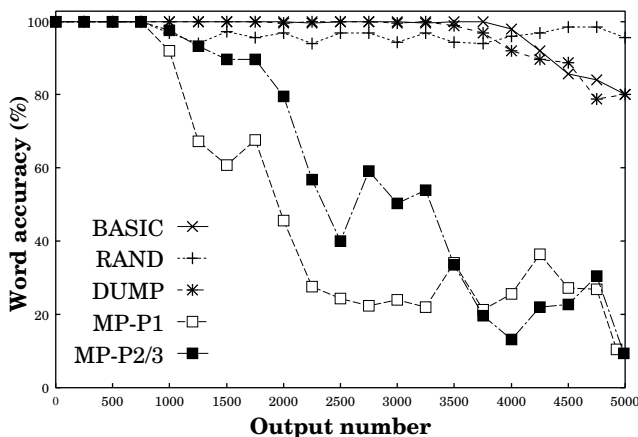

Figure 2: Progressive alignment accuracy

We can see a clear correlation between alignment cardinality and accuracy for the multi-pass classification method, and also a clear performance gain for the second pass over the first pass. The performance gain for the basic TF-IDF method over random sampling (word accuracy 96.64%) and the non-incremental method (word accuracy 96.20%) is more subtle, although the basic method does return higher word alignment accuracy and the output is more consistently accurate over the first 4000 or so annotated outputs, pointing to the success of the selective sampling method.

Conclusion

In conclusion, we have proposed two fundamentally different methods of unsupervised G-P alignment, and tested them on a set of 5000 Japanese G-P tuples. The first method centres around an adaptation of the TF-IDF met-

ric, and iterates over the data, hill-climbing as it goes. The second method, inspired by Ling and Wang (97), uses C4.5 to determine segment boundary compatibility for combined G-P context windows, and selects the most plausible overall alignment candidate according to the confident factor values returned by C4.5. It makes multiple passes over the data, incrementally enhancing alignment accuracy as it goes. The TF-IDF-based learning method returned a word accuracy of 96.94% in evaluation, surpassing the 58.18% 3-pass word accuracy for the C4.5 multi-pass classification method by a large margin. In the future, we are interested in running the different methods over data for other languages, and expanding evaluation.

References

Allen, J., Hunnicutt, M., and Klatt, D. (1987). *From Text to Speech: The MITTalk System*. CUP.

Baldwin, T. and Tanaka, H. (1999a). The applications of unsupervised learning to Japanese grapheme-phoneme alignment. In *Proc. of the ACL Workshop on Unsupervised Learning in Natural Language Processing*, pages 9–16.

Baldwin, T. and Tanaka, H. (1999b). Automated Japanese grapheme-phoneme alignment. In *Proc. of the International Conference on Cognitive Science*, pages 349–54.

Black, A., Lenzo, K., and Pagel, V. (1998). Issues in building general letter to sound rules. In *Proc. of the 3rd ESCA Workshop on Speech Synthesis*, pages 77–80.

Divay, M. and Vitale, A. (1997). Algorithms for grapheme-phoneme translation for English and French: Applications for database searches and speech synthesis. *Computational Linguistics*, 23(4):495–523.

Huang, C., Son-Bell, M., and Baggett, D. (1994). Generation of pronunciations from orthographies using transformation-based error-driven learning. In *Proc. of the International Conference on Speech and Language Processing*, pages 411–4.

Ling, C. and Wang, H. (1997). Alignment algorithms for learning to read aloud. In *Proc. of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97)*, pages 874–9.

Ling, C. and Zhang, B. (1998). Grapheme generation in learning to read English words. In Mercer and Neufield, editors, *Proc. of the 12th Biennial Conference of the Canadian Society for Computational Studies of Intelligence, AI'98*, pages 184–95. Springer.

Nagasawa, K., editor (1981). *Shinmeikai Dictionary*. Sanseido Publishers.

Quinlan, J. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Salton, G. and Buckley, C. (1990). Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4):288–97.

Sejnowski, T. and Rosenberg, C. (1987). Parallel networks that learn to pronounce English text. *Complex Systems*, 1:145–168.

Tsujimura, N. (1996). *An Introduction to Japanese Linguistics*. Blackwell.

Vance, T. (1987). *An Introduction to Japanese Phonology*. New York: SUNY Press.