In Proceedings of the International Conference on Cognitive Science, Tokyo, Japan, pp. 349-54.

Automated Japanese grapheme-phoneme alignment

Timothy Baldwin and Hozumi Tanaka
Tokyo Institute of Technology
{tim,tanaka}@cs.titech.ac.jp

Abstract

This paper describes an adapatation of the TF-IDF model to Japanese grapheme-phoneme alignment, without reliance on training data. The TF-IDF model is optionally complemented with affixation and conjugation handling modules, and determines frequencies through analysis of "alignment potential". The proposed system achieved a maximum accuracy of 94.74% on evaluation.

1 Introduction

The objective of this paper is to analyse the applicability of statistical methods to automated grapheme-phoneme alignment in Japanese, without reliance on pre-annotated training data. All possible grapheme-phoneme alignment mappings are generated exhaustively before being pruned through application of a series of constraints. Each individual alignment is then evaluated through a slightly modified form of the TF-IDF algorithm, optionally augmented with a basic model of adjective/verb conjugation and affixation.

1.1 Definitions/applications

Grapheme-phoneme ("G-P") alignment is defined as the task of *maximally* segmenting a grapheme compound into morpho-phonic units, and aligning each unit to the corresponding substring in the phoneme compound. For the purposes of this paper, **grapheme string** refers to the maximal kanji representation of a given word or compound, and **phoneme string** refers to the kana (hiragana and/or katakana) mora correlate. Strictly speaking, kana morae are made up of one or more consonant and a single vowel phoneme, except for the case of stand-alone vowels ([a], [i], [u], [e] and [o]) and consonants ([n]). However by using kana as our phonemic meta-unit, we are able to avoid consideration of phoneme combinatorial restrictions. Additionally, a unique broad phonetic transcription is associated with each kana character, such that the final step to full phonemic transcription can be achieved trivially. Another important thing to realise about kana characters is that they display a formative duality in that whereas they are essentially phonemic in nature, they can make their way into the grapheme string through phoneme replacement, or alternatively as particles, conjugating suffices or segments otherwise unrealisable by kanji. In this sense, 'grapheme' representation includes both kanji and kana where appropriate, whereas 'phoneme' representation is strictly limited to kana characters.

By **maximal segmentation** of the grapheme compound is meant that the grapheme string must be segmented to the degree that each segment corresponds to a self-contained component of the phonemic description of that compound, and that no segment can be further segmented into aligning sub-segments. To take the example of the grapheme string 感-謝-suru [*ka-n-sya-su-ru*] "to thank/be thankful",[1] 感 aligns with *ka-n* in the phoneme string, and 謝 with *sya*, as indicated in $align_1$ of Figure 1. The statement of 'maximality' of segmentation is qualified by the condition that each segment must constitute a morpho-phonic unit, in that for conjugating parts-of-speech, namely verbs and adjectives, the conjugating suffix must be contained in the same segment as the stem. For our verbal example of *ka-n-sya-su-ru*, however, the light verb status and conjugational independence of the suffix *su-ru* makes it a self-contained segment.

The main use of G-P alignment lies in its portrayal of phonological processes, and also implicit description of the range of readings each grapheme segment can take. We further suggest that a large-scale database of maximally aligned G-P tuples has applications within the more conventional task of G-P translation (Huang et al., 1994; Divay and Vitale, 1997).

1.2 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 phonetically. Clearly, the lower bound for atom size is a single character, but there is no inherent upper bound to the number of characters that can combine to form an atom.[2] 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 alignment of adjacent segments, and (iii)

[1] So as to make this paper as accessible as possible to readers not familiar with Japanese, kana characters are written italicised in Latin script throughout this paper, with character boundaries indicated by "-" and segment boundaries (which double as character boundaries) indicated by "⊙".

[2] In the context of a grapheme string, the upper bound on segment length becomes the character length of that string. Given that the maximum grapheme character length observed for the Shinmeikai dictionary was 9, this was the actual upper limit on segment length in evaluation.

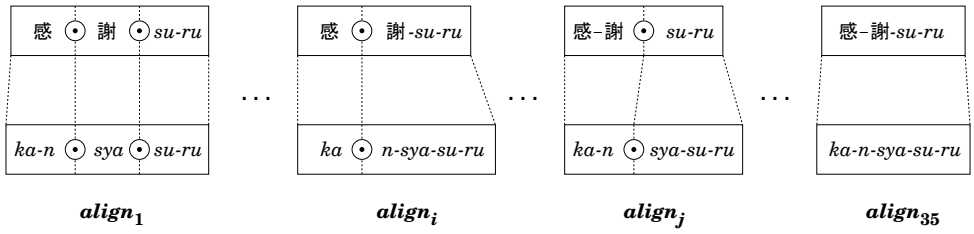

Figure 1: Candidate alignments for 感-謝-su-ru [*ka-n-sya-su-ru*] "to thank/be thankful"

phonetic similarity to regular readings in the case of novel G-P sub-alignment.

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 cognitive familiarity.

In the case that there is no alignment schema which produces familiar alignments for all individual grapheme segments, there appears to be 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 〈 白-髪〉–〈*si-ra-ga*〉, 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 upkeep the single known sub-alignment for 白 and produce a forced alignment for 髪, as in 〈 白 ⊙ 髪〉–〈*si-ra*⊙*ga*〉.

Finally, if a novel alignment must be made such as $\langle ...\odot$ 髪〉–$\langle ...\odot ga\rangle$ 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, or as is seen for 長-谷 [*ha-se*] in the surname 〈長-谷⊙川〉–〈*ha-se*⊙*ga-wa*〉.

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. All of these traits are inherent in our adaptation of the TF-IDF model.

The remainder of this paper is structured as follows. Section 2 describes the process of exhaustively generating all alignments between the grapheme and phoneme strings, and pruning off illegal alignments through a series of constraints. Section 3 then introduces the alignment scoring methodology, based on an adaptation of the TF-IDF algorithm optionally complemented with methods to handle conjugation and affixation. Finally, the proposed system is evaluated in Section 4.

2 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 alternation schemas and subsequent pruning through phonological constraints, and (c) scoring of all final candidate alignments to determine the final solution. We discuss the first two of these issues in this section, and devote Section 3 to discussion of the scoring mechanism.

2.1 Pre-processing

Lexical alternation is defined as the condition of there being multiple lexical spell-outs for a given phonetic content, all sharing the same basic semantics and kanji component. For Japanese, this can arise as a result of the replaceability of kanji with their corresponding kana (i.e. "*maze-gaki*"), or alternatively for *okurigana*, that is variation in kana suffices by way of phonetic content being conflated with or prised apart from the stem kanji phonetic content. An example of this latter process can be seen for the verb *ka-wa-ru* "to change", lexicalisable either as 変-*ru* or 変-*wa*-*ru*, with the underlined *wa* kana character conflating with the kanji stem of 変 in the former (basic) case for the same phonetic content.

Detection of **okurigana alternates** is achieved through analysing the graphemic form of G-P tuples sharing the same phonetic content, and aligning the graphemic component of each such corresponding tuple to determine kanji correspondence. All instances of okurigana-based lexical alternation are clustered together, and alternates of the 'basic' form removed from input. The basic form is defined as that with maximal phonemic conflation, that is minimal kana content. Note that this form of lexical alternation can extend over multiple kana units, and co-occur independently for individual kanji units.

The main purpose in clustering okurigana alternates is to enforce consistency of analysis over the alternation paradigm. This is achieved by applying the alternates in constraining the scope of alignment for the given alternation paradigm, by way of taking the intersection of the alignment paradigms for individual lexical alternates. The final alignment analysis for the base form is then applied to all alternates in forming our final combined output. That is, by removing all but the base form from final statistical analysis, we are guaranteed a unique alignment type, and at the same time can avoid having multiple realisations of the same basic item skew the performance of the statistical model.

2.2 Alignment of basic grapheme-phoneme tuples

The G-P alignment process can be subdivided into the three sub-tasks of (i) segmenting the grapheme string

into morpho-phonic units, (ii) aligning each grapheme segmentation to compatible segmentation(s) of the phoneme string, and (iii) pruning off illegal alignments through the application of a series of linguistic constraints.

Given that both grapheme and phoneme segments can be of arbitrary length, we must generate alignments for all cardinalities of segmentation. 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.

The first stage of the alignment process is thus to generate all possible segmentations of the grapheme string, by optionally placing a delimiter between adjacent characters (and implicitly placing delimiters at the beginning and end of both the grapheme and phoneme strings for all segmentation candidates). Note that individual kana and kanji characters are atomic, according to lexical constraint l:

$\langle l \rangle$ Segment boundaries can only exist at character boundaries. (*Characters are indivisible*)

Thus, 2^{m-1} segmentation candidates are generated for a grapheme string comprising m characters (both kana and kanji).

Next, the following axioms of alignment are applied in determining possible alignments for each segmentation paradigm.

$\langle a_1 \rangle$ The alignment must comprise a 1-to-1 function and, except in cases of "grapheme gapping", an isomorphism. (*no overlap in alignment and [conditionally] full* G-P *coverage*)

$\langle a_2 \rangle$ No crossing over of alignment is permitted. (*strict linearity of alignment*)

The conditionally isomorphic nature of grapheme-phoneme alignment gives rise to the property that delimiters in the phonemic string generally constitute phoneme segment boundaries, that is lead from one phoneme segment directly into the next. The crossing-over constraint further gives us the property that segments must be ordered identically in the grapheme and phoneme strings. Leaving "grapheme gapping" aside for the time being, the alignment process can thus be simplified by producing all segmentation candidates for the phoneme string, in an identical fashion as for the grapheme string, and returning the set product of all grapheme and phoneme segmentation candidates of equal cardinality. As for grapheme segmentation, phoneme segmentation produces 2^{n-1} candidates for a phoneme string n characters in length. Additionally, as the number of grapheme and phoneme segments must coincide for an isomorphic alignment, in actuality, we need generate segmentations up to a factor of only $l-1$, where $l = \min(m, n)$, giving the total number of alignments as $\sum_{x=0}^{l-1} C_x^{m-1} C_x^{n-1}$. Accordingly, 感-謝-su-ru [*ka-n-sya-su-ru*] is associated with 35 alignments, as suggested in Figure 1.

The exception to the isomorphism constraint is **grapheme gapping**, in which phoneme content is "gapped" from the grapheme string, such as can optionally occur for the *no* in 山-(<u>no</u>-) 手 [*ya-ma-<u>no</u>-te*] "uptown". Grapheme gapping only ever occurs for the phoneme segments *no* and *ga* and the head of geminates ("/Q/"), and a non-grapheme gapping alternate typically exists, as occurred for *ya-ma-no-te*. Note that alternation does not immediately point to grapheme gapping, as the *no* could potentially be conflating with 山 under alternation. As a result, we must fall back on the statistical model to weigh up the respective possibilities of conflation and grapheme gapping. At the same time, however, we want to be careful to play down the effects of grapheme gapping, due to its relative infrequence (0.1% in the solution set used in evaluation) and non-productive nature. Within the proposed formulation, this is achieved by considering grapheme gapping only in the case of appropriate lexical alternation.

The final step in alignment is to disallow any alignments which contravene the following linguistic constraints, applicable to grapheme segmentation ("G"), phoneme segmentation ("P"), and/or grapheme-phoneme alignment ("G-P"), respectively:

$\langle p_1 \rangle$ A demarkation in script form indicates a segment boundary, except for the case of kanji-hiragana boundaries. [G]

$\langle p_2 \rangle$ Graphemic kana must align with direct kana equivalents in the phoneme string. [G-P]

$\langle p_3 \rangle$ Intra-syllabic segments cannot exist for kana strings [G,P]

$\langle p_4 \rangle$ The length of a kanji substring must be equal to or less than the syllable length of the corresponding phoneme substring. [G-P]

$\langle p_5 \rangle$ Wherever possible, individual phoneme segments should contain a maximum of one voiced obstruent. [P]

Constraint p_1 produces the result that a segment boundary must exist at every changeover between hiragana and katakana, or kanji and katakana, and from hiragana to kanji. The exceptional treatment of kanji-hiragana changeovers is designed to facilitate the recognition of full verb and adjective morpho-phonic units, as these two parts-of-speech involve conjugating kana suffices and the potential for furigana-based lexical alternation. Indeed, if this allowance were not made, it would not be possible to apply a consistent analysis in the case of lexical alternation, as alternates involving hiragana suffices to the kanji stem would necessarily have the hiragana suffix segmented off from the preceding kanji (e.g. ⟨ 変 ⊙*<u>wa</u>-ru*⟩–⟨*ka*⊙*wa-ru*⟩, returning to our *ka-wa-ru* example, with the kanji stem corresponding to the *ka* phoneme), as would not be reproducible for the base form (i.e. ⟨ 変 ⊙*ru*⟩–⟨*ka-<u>wa</u>*⊙*ru*⟩). Note that p_1 is equally applicable to the grapheme and phoneme strings, and also applies to boundaries between Latin/Greek and Japanese scripts, as occurred for a handful of entries in evaluation.

Constraint p_2 polices the phonemic nature of kana, in disallowing alignment of kana segments of non-corresponding phonetic content. This has the double effect of pruning off peripheral alignments such as $align_j$ and, in combination with p_1, partitioning up grapheme strings which contain a 'pivotal' kana character sandwiched between kanji characters.

Constraint p_3, applicable to both grapheme and phoneme segmentation, introduces the notion that alignment operates on the *syllable*- rather than *character*-level.[3] While single kana characters generally function as individual syllables, stand-alone vowel and

[3]Syllable detection is based solely on consonant clustering (gemination and instances of [n]) and ignores stand-alone vowel kana as they produce syllable ambiguity, which we have no immediate way of resolving.

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

$$tf\text{-}idf(\langle g,p,ctxt\rangle) = \underbrace{\frac{freq(\langle g,p\rangle)-1+\alpha}{freq(\langle g\rangle)}}_{tf(\langle g,p\rangle)} \underbrace{\log\left(\frac{freq(\langle g,p\rangle)}{freq(\langle g,p,ctxt)-1+\alpha}\right)}_{idf(\langle g,p,ctxt\rangle)} \quad (2)$$

consonant kana (see above) can form syllable clusters with immediately preceding kana, as occurs for the *ka-n* combination in *ka-n-sya-su-ru*. Here, we would disallow a segment boundary to exist between *ka* and *n*, and as such prune off $align_i$ in Figure 1.

Constraint p_4 requires that each individual kanji character leads to a phoneme substring at least one syllable in length, irrespective of whether that single kanji comprises the head of a morpho-phonic unit or combines with adjoining kanji to form a multiple-grapheme segment. A two kanji segment is required, therefore, to align with a phoneme substring at least two syllables in length. p_4 provides surprising pruning potential for longer grapheme strings and is able to alleviate $align_j$ in Figure 1.

Finally, p_5 (a corollary of "Lyman's Law" (Vance, 1987, pp 136-9)) rejects multiple instances of voiced obstruents within a single phoneme segment, except where such exclusion would disallow all alignments. The defeasibility of this constraint is required because of the rare occurrence of alignments containing multiple voiced obstruents within a single phoneme segment, such as ⟨ 夥-*si-i*⟩–⟨*o-bi-ta-da-si-i*⟩ (voiced obstruents underlined). Despite p_5 occasionally producing over-constraint, it possesses significant disambiguating power in cases where a residue of legal alignment candidates is produced.

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

3 Alignment scoring

The scoring method utilised in this research is an adaptation of the TF-IDF model (Salton and Buckley, 1990), originally developed within the information retrieval ("IR") field for term weighting. The main point of departure in our usage of the TF-IDF model over the standard usage, is in the counting of frequencies.

The need for a special means of counting frequencies comes about because of the prevalence of G-P alignment paradigms displaying alignment ambiguity (i.e. full disambiguation is not achievable simply from the constraints of Section 2.2). When the system comes to count the frequency of a segment within a given context, it is unable to select that alignment candidate which is 'correct' from the containing alignment paradigm. Rather than attempting to pre-disambiguate the alignment paradigm to produce statistical absoluteness, the system takes each G-P tuple in turn and searches for the segment in question within its alignment paradigm; in the case that the target segment is indeed found to exist within one or more alignment candidates for the current G-P tuple, the system counts one. That is, frequency is based on *existence in the overall alignment paradigm* and not *actual occurrence in the final output*. In this way, we are able to process all G-P tuples in parallel and avoid having to commit ourselves to the plausibility of the system output for one G-P tuple over that for another.

3.1 Why TF-IDF?

Within the terms of the original IR-based application of the TF-IDF model, each grapheme segment type can be considered as a document, the associated phonemic segments across all G-P tuples as terms, and the left and right graphemic/phonemic contexts of the current grapheme/phoneme strings, as the document context.

The TF-IDF model maximally weights terms which occur frequently within a given document (TF) but relatively infrequently within other documents (IDF). As described in Section 1.2, we wish to maximally weight readings (aligned phoneme strings) which co-occur frequently with a given grapheme string, but at the same time score 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).

The optimisation of segment-level alignment is achieved by simultaneously considering the left and right contexts for both the grapheme and phoneme strings independently, and taking the average of the four resultant scores; this supports both the detection of regular alignments and a variable window size over the grapheme and phoneme strings. Additionally, by way of averaging the combined scores for each aligned segment to produce a single score for the overall alignment candidate, we are able to weight up alignments with more regularised segment-level readings, again mirroring the cognitive processing of G-P alignment.

While the 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 for the *ga* reading of 髪, in that they will generally be contained within the alignment paradigm of G-P tuples involving the grapheme segment in question.

3.2 Counting frequencies

The counting of term frequencies based on existence within an alignment candidate, can be formalised as in equation equation (1), in the case of $freq(\langle g,p\rangle)$. Here, p is the phoneme string aligning with grapheme string 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, 54-63) and gemination. We require some method to cluster frequencies for phonological alternates together so as to diminish the effects of data sparseness. Here, we are aided by the observation that phonological alternation affects only the phoneme string and

$$wtf(\langle g,p,ctxt\rangle) = \frac{freq(\langle g,p\rangle) - 1 + \alpha}{freq(\langle g\rangle)} \log\left(\frac{freq(\langle g,p\rangle)}{freq(\langle g,p\rangle) - freq(\langle g,p,ctxt\rangle) + \alpha}\right) \quad (3)$$

occurs only at the interface between adjacent segments, at the syllable level. By providing the system with a set of syllable-level phonological equivalence classes, it thus becomes possible to diagnose whether two phoneme segments are phonological alternates of one another and identify the more basic form.

In terms of the statistical formulation, this linguistic knowledge is applied as follows. For a given phoneme string $p = s_1 s_2 ... s_n$, where each s_i is a syllable unit, we generate a regular expression of all plausible phonological alternations $\{s_a|s_b|...\}s_2...\{s_\alpha|s_\beta|...\}$, where $\{s_a|s_b|...\}$ and $\{s_\alpha|s_\beta|...\}$ are the phonological equivalence classes for s_1 and s_n respectively. For example, given the phoneme string *ka-ku*, we would generate the string-level equivalence class $\{ka|ga\}\{ku|gu|\phi\}$,[4] where the *ka/ga* and *ku/gu* unit grapheme alternations are attributable to sequential voicing, and the *ku/*ϕ alternation to gemination.

The frequencies of all phonological alternations subsumed by the string-level equivalence class are then combined within $freq(\langle g,p\rangle)$. We are able to handle phonological alternation within the bounds of the original statistical formulation by virtue of the fact that the *grapheme* string is unchanged under phonological alternation, and as such the combined frequencies of alternates can never exceed the frequency of the associated grapheme string segment. This guarantees a *tf* value in the range $[0, 1]$.

3.3 The basic TF-IDF model

Our interpretation of the TF-IDF model is given in equation equation (2), where g is a grapheme unit, p a phoneme unit and *ctxt* some lexical context for $\langle g,p\rangle$ within the current alignment; $freq(\langle g\rangle)$, $freq(\langle g,p\rangle)$ and $freq(\langle g,p,ctxt\rangle)$ are the frequencies of occurrence of g, the tuple $\langle g,p\rangle$, and the tuple $\langle g,p\rangle$ in lexical context *ctxt*, respectively. The subtractions by a factor of one are designed to remove from calculation the single occurrences of $\langle g,p\rangle$ and $\langle g,p,ctxt\rangle$ in the current alignment, and α is an additive smoothing constant, where $0 < \alpha < 1$.

As described above, consideration of lexical context for a given 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, for both the left and right directions. In the case that $\langle g,p\rangle$ is a prefix of the overall G-P string pair, we disregard left lexical context outright and simply score according to the two right contexts. Correspondingly in the case of $\langle g,p\rangle$ being a suffix, we disregard right context. The resultant scores are then combined by taking the arithmetic mean. In the case of full-string unit alignment, the overall score is defined to be $tf(\langle g,p\rangle)$.

The overall score for the current alignment is determined by way of the arithmetic mean of the averaged scores for each segment pairing, with the exception of full kana-based grapheme segments which are removed from computation altogether.

3.4 Complementing the basic model

There are two commonly occurring segment types which do not fit in with the inherent 'high frequency, high disparity of context' philosophy of the basic TF-IDF model, namely affixation and verbal/adjectival conjugation.

Affixation refers to the condition of a grapheme segment with fixed reading commonly occurring as a prefix or suffix. In its basic form, the TF-IDF model as set out above weights down segments which occur in a fixed lexical context with high frequency, such that prevalent affixes could go undetected. To avoid this situation, we propose that the basic model be complemented with the "weighted term frequency" (*wtf*) metric as detailed in equation equation (3). This is applied for string-initial and string-final segments in parallel to the original *tf-idf* metric, and the maximum of the two resultant scores taken as the final score for the segment in question.

Verbal/adjectival conjugation is difficult to cope with given the existing statistical formulation, because it occurs concurrently at both the grapheme and phoneme levels (i.e. we have no immediate ceiling on combined frequencies as was the case for phonological alternation). As such, the existing model is not able to cluster frequencies for distinct conjugates together. We overcome the effects of conjugation-based alternation by pre-processing the input data to postulate verb paradigms,[5] based on conjugational analysis of the kana suffix to a given stem (Baldwin, 1998). This process is performed independently of any verb dictionary, and is based simply on a search for segments comprising a stem and inflectional suffix; such segments are classified according to the invariant stem content and each conjugational class which could have produced the given inflection. Segments sharing a common stem and conjugational class are then clustered together to form our dynamic verb paradigms, and further expanded through analysis of phonological alternation as described above.

Scoring of verb conjugates is facilitated by counting the number of occurrences of the given verb stem in the full set of alignment paradigms (as either a fully or partially aligned segment), to take as our $freq(\langle g\rangle)$. $freq(\langle g,p\rangle)$ is then set to the combined frequency of occurrence of all conjugates belonging to the current verb paradigm. Given that conjugational analysis inherently delimits both the grapheme and phoneme segments, we have no further use for lexical context, and hence $freq(\langle g,p,ctxt\rangle)$ is set to $freq(\langle g,p\rangle)$, and the respective values plugged into equation equation (3).

An additional extension to the basic model based on incremental learning is proposed in (Baldwin and Tanaka, 1999).

4 Evaluation

The proposed system was tested on a set of 59744 G-P tuples containing at least one kanji, derived from the combined EDICT Japanese-English[6] and Shinmeikai (Nagasawa, 1981) dictionaries. The makeup of this input set is given below:

No. of entries (+ lex. alternations): 59744
No. of entries (− lex. alternations): 51484
Ave. grapheme string char. length: 2.35

[4] Here, ϕ designates the head of a long consonant, also indicated by /Q/ in phonological theory.

[5] Although discussion hereon refers exclusively to verbs, (conjugating) adjectives are handled in exactly the same manner.

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

Ave. kanji in grapheme string: 1.93
Ave. phoneme string char. length: 3.98
Ave. phoneme string syllable length: 3.46

Due to restrictions on manpower, it was unfeasible to fully annotate all G-P tuples, and a limited set of 5000 tuples was instead randomly selected for manual annotation. Given the direct and indirect statistical interaction between these 5000 tuples and the remainder of the dictionary, however, the behaviour of the system in this restricted evaluation is suggested to be indicative of overall performance. By way of note, the average number of (grapheme) segments for the random set of 5000 G-P tuples was 1.95.

First, looking to the relative applicability of the proposed constraints, we evaluated the effectivity of alignment pruning. The average number of exhaustively generated alignment candidates, prior to the application of the constraints, was 15.57, and the average residue of alignment candidates after constraint application was 4.07, a reduction of nearly 75%. For the 5000 member evaluation set, there were no cases of over-constraint, such that the solution was always contained in the final alignment paradigm for each G-P tuple. It is also worthwhile noting that full disambiguation was produced for 895 G-P tuples in the evaluation set.

Turning now to alignment accuracy, we tested the proposed methodology with differing values of α, and affix and conjugation handling variously activated and deactivated. This produced the results given in Figure 2, with each curve representing a different value of α and the various affix ($\pm$A) and conjugation ($\pm$C) handling activation states indicated on the x-axis. Bilac et al. (1999) provide a baseline for evaluation, by way of a rule-based formulation which achieved a 92.90% accuracy on the same dictionary file.

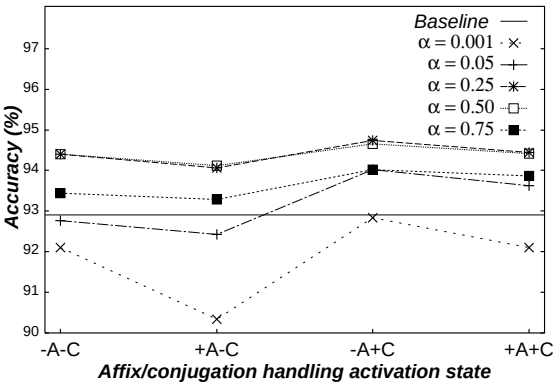

Figure 2: Alignment accuracy

From Figure 2, it would appear that the optimal system performance is achieved with α set to a value between 0.25 and 0.5, with the best single accuracy of 94.74% achieved with α set to 0.25, affix handling off and conjugation handling on; this represents a significant improvement over the baseline figure of 92.90%. Interestingly, affix handling seems to marginally diminish system performance, but conjugation handling improves it perceptibly, irrespective of the value of α. Conjugation handling appears to reduce "over-alignment" (proper subsumption of the correct alignment by the system alignment) of conjugating atoms, while not generating significant levels of underalignment (proper subsumption of the system alignment by the correct alignment). Affix handling acts similarly in not affecting underalignment, but at the same time generates additional overalignment errors. Interestingly, the adverse effects of affix handling are generally restricted to non-conjugating atoms, such that when conjugation and affix handling are combined, the advantages of conjugation handling are retained, although unwanted noise is still produced by the affix handling. This suggests that higher performance could be expected given a more successful affix handling method.

Increasing the value of α produces progressively lower levels of overalignment and local decreases in underalignment, bottoming out around 0.25, from which point onward, any decreases in overalignment were outweighed by increases in underalignment. These results contrast sharply with those presented in (Baldwin and Tanaka, 1999) for the same basic model, taking an augmented version of the 5000 member evaluation set as input. In (Baldwin and Tanaka, 1999), the system accuracy increased monotonically as α tended toward 0, suggesting that smaller values of α may be less susceptible to the effects of data sparseness.

5 Conclusion

In this paper, we proposed an adaption of the TF-IDF model to Japanese grapheme-phoneme alignment, and further went on to develop modules for handling the effects of affixation and conjugation. Our implementation of the TF-IDF model is characterised by it not requiring training, and instead sourcing non-disambiguated alignment paradigms to determine the alignment potential of a given segment or tuple. Evaluation indicated that our proposed method was able to outperform a rule-based system on a common test set and that conjugation handling significantly enhanced the system accuracy, although affixation handling proved detrimental to performance and remains an issue for further research.

References

T. Baldwin and H. Tanaka. 1999. 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.

T. Baldwin. 1998. *The Analysis of Japanese Relative Clauses*. Master's thesis, Tokyo Institute of Technology.

S. Bilac, T. Baldwin, and H. Tanaka. 1999. Incremental Japanese grapheme-phoneme alignment. In *Information Processing Society of Japan SIG Notes*, volume 99-NL-209, pages 47–54.

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

C.B. Huang, M.A. Son-Bell, and D.M. Baggett. 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.

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

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

N. Tsujimura. 1996. *An Introduction to Japanese Linguistics*. Blackwell.

T.J. Vance. 1987. *An Introduction to Japanese Phonology*. New York: SUNY Press.