# Automatically Generating Related Queries in Japanese

ROSIE JONES, KEVIN BARTZ, PERO SUBASIC and BENJAMIN REY
*Yahoo Inc., 3333 Empire Ave, Burbank, CA 91504, USA*
*Email: { jonesr, benjamir}@yahoo-inc.com, bartz@fas.harvard.edu*, pero@acm.org**

**Abstract.** Web searchers reformulate their queries, as they adapt to search engine behavior, learn more about a topic, or simply correct typing errors. Automatic query rewriting can help user web search, by augmenting a user's query, or replacing the query with one likely to retrieve better results. One example of query-rewriting is spell-correction. We may also be interested in changing words to synonyms or other related terms. For Japanese, the opportunities for improving results are greater than for languages with a single character set, since documents may be written in multiple character sets, and a user may express the same meaning using different character sets. We give a description of the characteristics of Japanese search query logs and manual query reformulations carried out by Japanese web searchers. We use characteristics of Japanese query reformulations to extend previous work on automatic query rewriting in English, taking into account the Japanese writing system. We introduce several new features for building models resulting from this difference and discuss their impact on automatic query rewriting. We also examine enhancements in the form of rules which block conversion between some character sets, to address Japanese homophones. The precision/recall curves show significant improvement with the new feature set and blocking rules, and are often better than the English counterpart.

**Key words:** kanji in web search, Japanese web search queries, query processing, query substitution, query reformulation

## 1 Introduction

Because Japanese has three writing systems, Japanese web search queries can often be written in several forms with equivalent meanings. Queries can also be expressed partially or completely using Roman letters for English words or other foreign words, celebrities and brand names. These same queries could be expressed using *katakana,* the special Japanese writing system for foreign words. Thus a pair of queries can be equivalent in meaning, but appear different to a language-agnostic search engine.

Similar problems exist with information retrieval systems in other languages, which are often unable to retrieve documents due to a difference in vocabulary choice. For example, a user issues the query "*cat cancer,*" but all documents in the collection use the expression "*feline cancer.*" In addition, a user's search query can be an imperfect description of their information need, and automatic reformulation can help the user better express that need. Existing solutions to these problems include relevance feedback (Salton and Buckley, 1990) and pseudo-relevance feedback, query term deletion (Jones and Fain, 2003), and substituting query terms with related terms from retrieved documents (Terra and Clarke, 2004).

Pseudo-relevance feedback involves submitting a query for an initial retrieval, processing the resulting documents, modifying the query by expanding it with additional terms from the documents retrieved and then performing a second retrieval with the modified query. Pseudo-relevance feedback has limita-

[*] This work was done while Kevin Bartz and Pero Subasic were employees at Yahoo!.

tions in effectiveness (Ruthven, 2003). It may lead to query drift, as unrelated terms are added to the query. It is also computationally expensive. Substituting query terms with related terms from retrieved documents also relies on an initial retrieval. Query relaxation or deleting query terms leads to a loss of specificity from the original query. Our setting is sponsored search, in which we attempt to match enormous numbers of queries to a much smaller corpus of advertiser listings. Here recall is crucial, as conjunctive search often leads to no matching results.

Automatic query rewriting was first examined by Jones et al. (2006). Mining user query sessions, in which web searchers modify their queries with related terms, allows us to build a collection of related queries and phrases. Given a new query, we can generate related queries by choosing from the database of related queries, or by breaking it into phrases and substituting individual phrases. Jones et al. use machine learning to automatically learn a model to identify the most related rewrites. Their learned model showed that the best rewrites are generated by finding query reformulations which have small character edit distance and many words in common. When phrase-level substitutions are used, it is better to change as few phrases as possible. This method gives promising results for English. When applying it to other languages we may need to take into account features of the writing system of the target language. For example, a query written in *kanji* (the pictograms derived from Chinese) may have completely different characters than the same words written in *hiragana* (the Japanese syllabary) and so appear to have high character edit distance.

In Section 2 we look at the typical use of Japanese writing systems in Japanese query logs. In Section 3 we give a high-level overview of the existing approach for generating related queries. In Section 4 we introduce web search query session substitutable pairs as one resource for substitution generation; the second resource, phrase-level substitutables are introduced in Section 5. In Section 6 we discuss character normalization approach over multiple character sets. Section 7 explains how to combine and rank query and phrase level substitutables. In Section 8 we give results for our Japanese-language-specific scoring function and show that it improves on a language-independent system.

## 2 Use of Japanese character sets in Web search query logs

Japanese queries typically consist of a combination of *kanji*, *hiragana*, *katakana* and *romaji* (Latin or Roman characters). **Table 1** shows examples of each of these character types, along with how many are found on average per query. Kanji, Chinese characters used in Japanese writing, are the main carriers of semantics in Japanese texts: kanji compounds are used to build nouns. Hiragana is a phonetic syllabary of 48 basic characters used to write grammatical markers and endings. In modern use, hiragana is used often instead of kanji when the meaning is unambiguous from the context. Combinations of kanji and hiragana are used for verbs and adjectives. Katakana is a phonetic syllabary of 48 basic characters, corresponding to the same sounds as the hiragana characters, used to write foreign and loan words. However, today katakana is quite often used for Japanese personal names, brand names, even for words that are normally written in kanji. The Roman alphabet is also used for foreign names, loan words, mathematic and scientific notation and so on. A small number of queries contain numerals or special symbols such as the wave (～) or dot ( ・ ).

| **Character Type** | **Average Per Query** | **Example** |
|---|---|---|
| Kanji | 2.49 | 車 |
| Hiragana | 0.57 | は |
| Katakana | 2.69 | ト |
| Roman | 1.86 | A |
| Space | 0.522 | |
| Special | 0.00534 | ・ |

**Table 1. Average characters of each type in Japanese web search queries.**

Table 2 shows a breakdown of queries based on the character sets used in them, estimated from a sample of nearly 100 million Web search queries from Japanese query logs. Surprisingly, spaces are used in 38.4% of queries, despite being exceedingly rare in Japanese newspaper text. Another difference is the amount of kanji: only 2.5 characters, or 30.9%, of the average Japanese query are kanji (see **Table 1**). This is a significantly smaller proportion than the 43% found in newspaper text (Chikamatsu et al., 2006).

| **Query Property** | **Number in Sample** | **%** |
|---|---:|---|
| Total | 96,557,021 | 100% |
| Unique | 19,902,238 | - |
| **Containing Kanji** | **60,946,078** | **63.1%** |
| **Containing Hiragana** | **18,070,419** | **18.7%** |
| **Containing Katakana** | **44,111,274** | **45.7%** |
| Containing Roman | 21,779,928 | 22.6% |
| Containing spaces | 37,116,974 | 38.4% |
| Containing special characters | 515,266 | 0.5% |
| Containing Roman and Kanji | 6,689,319 | 6.9% |
| Containing Roman and Hiragana | 1,649,224 | 1.7% |
| Containing Roman and Katakana | 5,242,453 | 5.4% |
| Containing Hiragana and Kanji | 1,3963,128 | 14.5% |
| Containing Hiragana and Katakana | 6,681,077 | 6.9% |
| Containing Katakana and Kanji | 23,814,348 | 24.7% |

**Table 2. Distribution of character types in Japanese queries.**

## 3 Generating related queries

In this section we give a high-level overview of the general approach we use for generating related queries. We go into detail in subsequent sections. Jones et al. (2006) generate related queries based on query reformulation from user sessions in web search query logs. The overall process consists of the following steps: (1) Generate database of related queries and phrases by data mining from query logs (details in Sections 4 and 5). (2) Build a model of good rewrites using machine learning from hand-labeled examples. (3) Generate related queries for incoming queries using databases from step (1) and score using the similarity function learned in step (2).

We use machine learning to generate a scoring function for identifying high-quality rewrites. The overall process consists of taking a sample of queries, generating several rewrites for each, then having the rewrites scored for quality. We then extract features for the rewrite pairs, and use machine learning to identify which features are the strongest indicators of high-quality rewrites. We will see more details about the features we consider in Section 7.

## 4 Query session substitutable pairs

To generate related queries (Jones et al., 2006), we look to user query sessions in a search engine query log to find related phrases. Users often modify their queries in a search session (Jones and Fain, 2003; Spink and Jansen, 2004). These modified queries may contain related queries and phrases that we can use for query generation. In Table 3 we see a breakdown of the relationships in sequential pairs of Japanese queries, based on manual labeling of a random sample of 100 sequential query pairs. The rate of word deletion is similar to that found by (Jones and Fain, 2003) on US English queries, while the insertion rate is twice as high.

### 4.1 FILTERING QUERY SESSION SUBSTITUTABLES

In order to remove the unrelated query pairs we saw in . Table 3, we run a significance test on all candidate query pairs from a period containing 96 million search queries. The significance test we use is the log-likelihood ratio test (Manning and Schuetze, 1999), which tests whether P(query 2 | query 1) >> P(query 2) at a given level of significance. Here we take P(query 2) to be the overall probability of a user searching on query 2, and we treat P(query 2 | query 1) as the probability that a user issues query 2 immediately after issuing query 1. Of the three million unique query pairs from our period, only 1.2% passed this test. Table 4 shows the distribution of rewrite types by class for a sample of 100 query pairs passing the statistical test. Only 9% of pairs are unrelated, showing the test to be effective at identifying semantically related pairs.

| Rewrite Type | Examples | % |
|---|---|---|
| No relationship | ２ｃｈ (*2 ch*; a popular portal) → zippo | 48% |
| Word insertion | ２ｃｈ→ ２ｃｈ website | 20% |
| Word substitution | 日本放送 (*nippon housou*; Japan broadcasting) →日本テレビ (*nippon terebi*; Japanese television) | 9% |
| Word deletion | 浮舟 ダウンロード ギタドラ (*ukifune daunroodo gita dora*; Ukifune guitar drama download) → 浮舟 ギタドラ (*ukifune gita dora*; Ukifune guitar drama) | 6% |
| Spelling change | ブールス・ウィルス (*buurusu wirusu*; Bruce Willis) → ブールス・ウイルス (*buurusu uirusu*; Bruce Willis) | 2% |
| Non-substantive change (spacing, encoding) | 居宅 介護 (*wakashi taku*; Wakashi nursing home) → 居宅介護 (same except for space) | 1% |
| Homophone switch | 幕府website (*bakufu website*; shogunate Web site) → bakufu website | 5% |
| Related meaning | ドコモ (*docomo*; NTT mobile phone service) → au (“au” mobile phone service) | 9% |

**Table 3. Breakdown of rewriting types for a random sample of 100 sequential query pairs.**

| Rewrite Type | % |
|---|---|
| Unrelated | 9% |
| Word insertion | 41% |
| Word substitution | 5% |
| Word deletion | 6% |
| Spelling change | 1% |
| Non-substantive change | 9% |
| Homophone switch | 2% |
| Related meaning | 27% |

**Table 4. Breakdown of substitutable types for query pairs with likelihood ratios greater than 70.**

## 4.2 MANUAL LABELING OF QUERY SESSION SUBSTITUTABLES

In order to quantify the quality of query rewrite suggestions from query-session substitutables, we used a measure from (Jones et al., 2006): “1” for an unambiguous misspelling or semantically equivalent synonym; “2” for a generalization or specification; “3” for a sibling or broad match; and “4” for an unrelated suggestion.

| Query 1 | Query 2 | Label | % |
|---|---|---|---|
| 新車保険 (*shinsha hoken*; new car insurance) | 車保険 (*kuruma hoken*; car insurance) | 1 | 13.0% |
| この指止まり (*kono yubi tomari*; a popular phrase) | この指止まれ (*kono yubi tomare*; a popular phrase) | 1 | |
| アップルMP３プレイヤ ー (*appuru MP3 pureiyaa*; Apple MP3 player) | Ipod | 2 | 41.5% |
| ランドセル (*randoseru*; knapsack) | カバン (*kaban*; bag) | 2 | |
| めがね (*megane*; glasses) | コンタクトレンズ (*kontakuto renzu*; contact lenses) | 3 | 39.3% |
| 木村拓哉 (*kimura takuya*; Kimura Takuya – Japanese celebrity) | 稲垣五郎 (*inagari gorou*; Inagari Gorou – Japanese celebrity) | 3 | |
| このゆびとまれ (*kono yubi tomare*; a popular phrase) | ２ｃｈ (*2ch*; a popular Web portal) | 4 | 6.2% |

**Table 5. Editorial scoring examples.**

To test our method’s performance under this metric, we first ran the log-likelihood test on a large data set containing 1.5 billion queries. We then sampled 1,000 queries at random and produced for each the two suggestions with the highest log-likelihood ratios. Finally, a content editor scored the suggestions under the (Jones et al., 2006) metric. The results are in line with our previous evaluation, showing that query-session substitutables most often score 1 or 2. Table 5 shows a breakdown of the results with examples.

Although they are highly relevant, one weakness of query-session substitutables is their relatively poor *coverage* of the general query space. In the context of a Web search engine, we define coverage as the percentage of user-entered queries for which our rewriting system produces a suggestion. To estimate coverage, we sampled 100,000 queries from search logs from a week after the period used to generate the substitutable pairs. Query-session substitutables generated a suggestion for just 39.3% of these 100,000 queries. When we exclude adult substitutables and those that remove a company's trademark, coverage drops to 33.1%. In the next section we develop a method to expand our system's coverage.

## 5 Phrase-level substitutables

We may be able to improve coverage by finding phrases within the query and replacing them with appropriate phrases. In this section we describe an approach for identifying phrases within queries and generating new queries using those phrases.

### 5.1 PHRASE IDENTIFICATION IN SEARCH QUERIES

In segmenting Japanese queries, we would like the freedom to exchange individual nouns and verbs without modifying any grammatical particles. Thus we do not use the *bunsetsu* approach (Makino and Kizawa 1980), which leaves particles attached to the nouns and verbs they modify. We instead apply a proprietary segmentation technique from BasisTech, a Japanese-language morphological analyzer, which isolates both grammatical particles and content words. For example, BasisTech breaks "USBポートに挿す" (*USB pooto ni sasu*; to insert into a USB port) into "USB ポート | に | 挿す |" (*USB pooto* | *ni* | *sasu*; to insert | into | a USB port). We found Japanese Web search queries to contain an average 2.9 tokens. This is close to the 2.8 tokens typically found in English-language queries (eg Spink and Jansen, 2004).

We next group these segments into *units* by identifying adjacent tokens with high mutual information (Kapur and Parikh, 2006). For example, BasisTech segments 電車男 (*densha otoko*; train man) into 電車 | 男 (*densha* | *otoko;* train | man), but since this is the name of a popular television show that occurs frequently in Web search query logs, the mutual information measure identifies it as a single phrase. Some examples of queries with both segmentations and phrase groupings are shown in Table 6.

| Query | Segmentation | Phrase Grouping |
|---|---|---|
| 電車男 (*densha otoko*; train man) | 電車 \| 男 (*densha* \| *otoko*; train \| man) | 電車 男 (*densha otoko*; train man) |
| 電車男番組 (*densha otoko bangumi*; train man show) | 電車 \| 男 \| 番組 (*densha* \| *otoko* \| *bangumi*; train \| man \| show) | 電車 男 \| 番組 (*densha otoko* \| *bangumi*; train man \| show) |
| 福岡のキャナルシティー (*fukuoka no kyanaru shitii*; Canal City in Fukuoka) | 福岡 \| の \| キャナル \| シティー (*fukuoka* \| *no* \| *kyanaru* \| *shitii*; Canal \| City \| in \| Fukuoka) | 福岡 \| の \| キャナル シティー (*fukuoka* \| *no* \| *kyanaru shitii*; Canal City \| in \| Fukuoka) |

**Table 6. Segmentation and unitization for sample Japanese queries.**

### 5.2 IDENTIFYING PHRASE SUBSTITUTABLES FROM SEARCH SESSIONS

Many sequential search queries have a single phrase substituted. For example, a user might type 福岡のキャナルシティー (*fukuoka no kyanaru shitii*; Canal City in Fukuoka) and then rewrite it, changing 福岡 (*fukuoka*; Fukuoka) to 福岡県 (*fukuoka ken*; Fukuoka Prefecture) and re-submit the query. Over a large data set of 1.5 billion queries, we identified sequential query pairs with a single phrase substituted and collected the substituted phrases into a database of phrase substitutables.

We then filtered phrase substitutables using the same log-likelihood test we used to filter whole-query substitutables, which led to a 98.3% drop in the number of unique phrase substitutions.

## 5.3 GENERATING QUERY SUGGESTIONS USING PHRASE SUBSTITUTABLES

When our system receives an input query, we first segment it into phrases with BasisTech and regroup the result into phrases using mutual information, as described in Section 5.1. We then look up the top-scoring substitutables for each individual phrase. To produce a suggestion candidate, we swap at most two phrases in any given query with each of its substitutable phrases. Our system thus builds a combinatorial set of phrase-substituted suggestions.

It is possible for some of the candidates to be nonsensical when the units are imperfect. For instance, substituting 船渠 (*senkyo*; dock) for キャナル (*kyanaru*; canal) in the query 福岡のキャナルシティー (*fukuoka no kyanaru shitii*; Canal City in Fukuoka) results in a nonsensical suggestion. As a simple sanity check, we accept only phrase-substituted suggestions that are Yahoo! Search Marketing bidded terms. This means that an advertiser has placed a pay-per-click ad on a particular term, which is a good sign that the term makes sense.

When we add phrase-level substitutables to query-level substitutables, coverage (the percent of queries for which we are able to generate a rewrite from our databases) increases from 33.1% to 43.6% of search volume.

## 5.4 EVALUATION OF SUGGESTIONS FROM PHRASE SUBSTITUTABLES

We separately evaluated phrase substitutables, using the same one-to-four rubric as for the whole queries. Starting from the 1,583 input queries for which we generated query-level substitutables, we generated query suggestions using the methodology described in Section 5. We then selected two at random for each query to send to a content editor for scoring. The results in Table 7 below show that phrase-level substitutables are even more relevant than query-level substitutables. Although the 1,583 input queries used in this evaluation were not the same as the 1,000 queries used in 4.2, both input sets were chosen at random, so the comparison is valid.

| Score | % in query-level | % in phrase-level |
|---|---|---|
| 1 | 13.0% | 24.8% |
| 2 | 41.5% | 25.2% |
| 3 | 39.3% | 44.5% |
| 4 | 6.2% | 5.5% |

**Table 7. Results of phrase substitutables evaluation and comparison with query substitutables.**

## 6 Japanese query character normalization over multiple character sets

To define edit-distance features which take into account the multiple character-sets used in Japanese, we may wish to consider first normalizing Japanese text to a single writing system. We consider three possible normalizations and in Table 8 show them for the sample query テレビ番組表 CBC (*terebi bangumi hyou CBC*; CBC television schedule).

| Input Form | Description | Example |
|---|---|---|
| raw form | query as entered by the user | テレビ番組表 CBC |
| Kanji-only form | query’s kanji part | 番組表 |
| Romaji form | query after conversion to romaji | terebi bangumi hyou CBC |

**Table 8. The query "CBC television schedule" under different character-set normalizations.**

The *raw form* of a query is the set of characters chosen and entered by the user. An advantage of this representation is that we can be sure we have not lost any of the user's meaning through transformations. This is the form we would use if we applied a model based on English language data to Japanese without any modification.

We define a query's *kanji-only form* as the ordered set of kanji contained in the query. Since these characters make up content words – kanji do not have any grammatical function – we expect it to contain a query's most significant concepts. In the context of a rewriting system, query pairs containing common kanji could be considered to have preserved most of the core meaning. As we saw in Table 2, 36.9% of queries contain no kanji, so similarity based on this form will be useful for only some query rewrite pairs.

To obtain *romaji form*, we first convert all kanji to kana using the open-source Kakasi software, which segments a query and assigns each word the corresponding kanji. We then convert the kana to ASCII using the Revised Hepburn Romanization system (ANSI, 1972). After performing this conversion on a sample of 100,000 queries, we find an average 10.7 romaji characters per query. Note that while romaji form is equivalent to a normalized kana form for queries with no roman characters, it differs for queries containing roman characters (22.6% of queries, as shown in Table 2).

## 7 Combining and ranking query and phrase level suggestions

In Section 4 we described a method for generating query suggestions based on whole queries. In Section 5 we described a method for generating suggestions based on phrase substitutions. We now look at a way of combining these and ranking them, based on taking advantage of the multiple writing forms in Japanese. To do so, we define a set of features based on edit distance and statistical information. Then we fit a linear model to predict the quality of the suggestion, and use this score to rank the suggestion candidates.

### 7.1 EDIT DISTANCE AND STATISTICAL FEATURES

We considered a variety of edit distance measures to detect pairs whose substitutability is explained by changes in writing system. We applied Levenshtein distance to each of the normalized Japanese writing forms described in Section 6. Furthermore, to assess the similarity of the kanji parts of queries, we also calculated the *kanji disagreement* as the percentage of kanji not shared by the two queries. In Table 9 we show sample query pairs that we would like to recognize as high-quality rewrites. Next to each pair is the edit distance measure designed to detect the similarity at hand.

We also used some variants on those features, as well as other lexical indicators:

- Romaji Levenshtein after removing spaces (*levrs*)
- Levenshtein distance between the queries' kana (*levk*)
- prefix overlap in Romaji characters normalized by the query length (*opr*),
- Boolean valued one when a digit within the query is modified (*digit*)
- Boolean valued one when a Japanese character is present at all (*japanese*).

| **Query 1** | **Query 2** | **Measure** | **Value** |
|---|---|---|---|
| インシュランス (*in-shuransu*; insurance) | インシュアランス (*in-shuaransu*; insurance) | Levenshtein raw form | 0.125 |
| ういんず (*uinzu*; winds) | ウインズ (*uinzu*; winds) | Levenshtein romaji (*levr*) | 0.000 |
| 七五三 写真 (*shichi go san shashin*; shrine festival photos) | 七五三 写真館 (*shichi go san shashinkan*; shrine festival photo studio) | Kanji disagreement (*kanjid*) | 0.166 |

**Table 9. Edit distances defined over different forms and normalizations of query pairs.**

We also considered a number of features related to the statistics of the substitution chosen. These include:

- Likelihood ratio of the substitution
- Frequency of the substitution
- Probability of the substitution (minimum where multiple phrases are substituted)
- Mutual information of the substitution

## 7.2 INFORMATION GAIN

Using the scored query pairs in Sections 4.2 and 5.4, we computed the information gain provided by each class of features. Romaji Levenshtein distance proved to be the best discriminator, perhaps because of its robustness in detecting equivalent text across all Japanese writing methods. Forms of edit distance using kanji and raw forms were somewhat less useful because they were less able to detect similarities across writing types. The example ホンダ (*honda*; honda) → honda, illustrates a class of equivalence in which only the romaji edit distance is helpful. In Table 10 below, a score of 100% means that the variable provides a perfect reordering, corresponding to a 100% drop in entropy of the training set.

| | |
|---|---|
| Levenshtein distance of Romaji (spaces removed) (*levrs)* | 25.5% |
| Levenshtein distance of Romaji (with spaces) (*levr)* | 24.7% |
| Jaccard distance of Romaji words (*wordr)* | 23.9% |
| Prefix overlap of Romaji (*opr*) | 22.6% |
| Levenshtein distance of surface form (*lev*) | 17.1% |
| Levenshtein distance of kana *(levk)* | 16.9% |
| Probability of substitution (*p12min*) | 11.4% |
| Jaccard distance of surface form words (*wordr*) | 7.3% |
| Kanji disagreement (*kanjid*) | 6.1% |
| Binary variable for presence of Japanese characters (*japanese)* | 4.4% |
| Number of substitutions made (*numSubst*) | 4.3% |
| Whether a digit change took place (*digit*) | 1.1% |

**Table 10. Information gain provided by each feature in discriminating rewrite pairs' scores.**

## 7.3 RANKING FUNCTION

Using the random pairs, we used stepwise forward and backward linear regression to find the best linear model to predict the one-to-four score defined in Section 4.2. The feature set included all variables in Table 10. The model selected is shown below. Interestingly, the bulk of the score is derived from edit distance features. The only statistical feature which made it through the best subset is the probability of substitution. For US queries the same effect has been observed (Jones et al., 2006).

$$\begin{aligned} LMScore(q,q') = \ & 1.34371 - 1.13609 \times levr(q,q') + 1.97118 \times levrs(q,q') + 0.46919 \times wordr(q,q') \\ & + 0.49280 \times digit(q,q') + 0.24153 \times kanjid(q,q') - 0.37652 \times opr(q,q') \\ & + 0.09991 \times japanese(q,q') - 0.25832 \times levk(q,q') - 0.21648 \times p12\min(q,q') \end{aligned}$$

## 7.4 PERFORMANCE COMPARISONS

We next compare our 10-feature model to simpler functions inspired by other recent literature on query-session substitutables. Jones et al. (2006) fit a model using only Levenshtein distance, Jaccard distance of the query pairs' words and the number of substitutions made.

We apply Jones et al.'s model to our data twice, once using the queries' raw forms and once with their romaji forms. We then compare it to the precision of our 10-feature learned function. Holding the folds constant, we apply 10-fold cross-validation, fitting a model for each fold using each of the three feature sets under comparison, and report precision and recall. In Figure 1 below, the horizontal line indicates that the baseline proportion of good rewrites without ranking is 51%. The solid black line denotes the performance of a direct application of (Jones at al)'s model to Japanese substitutable pairs. The dashed line represents the performance of the same model, with features computed after converting all Japanese characters to romaji. We see that this collapsing of Japanese character sets provides the biggest boost, with only a small gain from augmenting this feature set from three to 10 features.

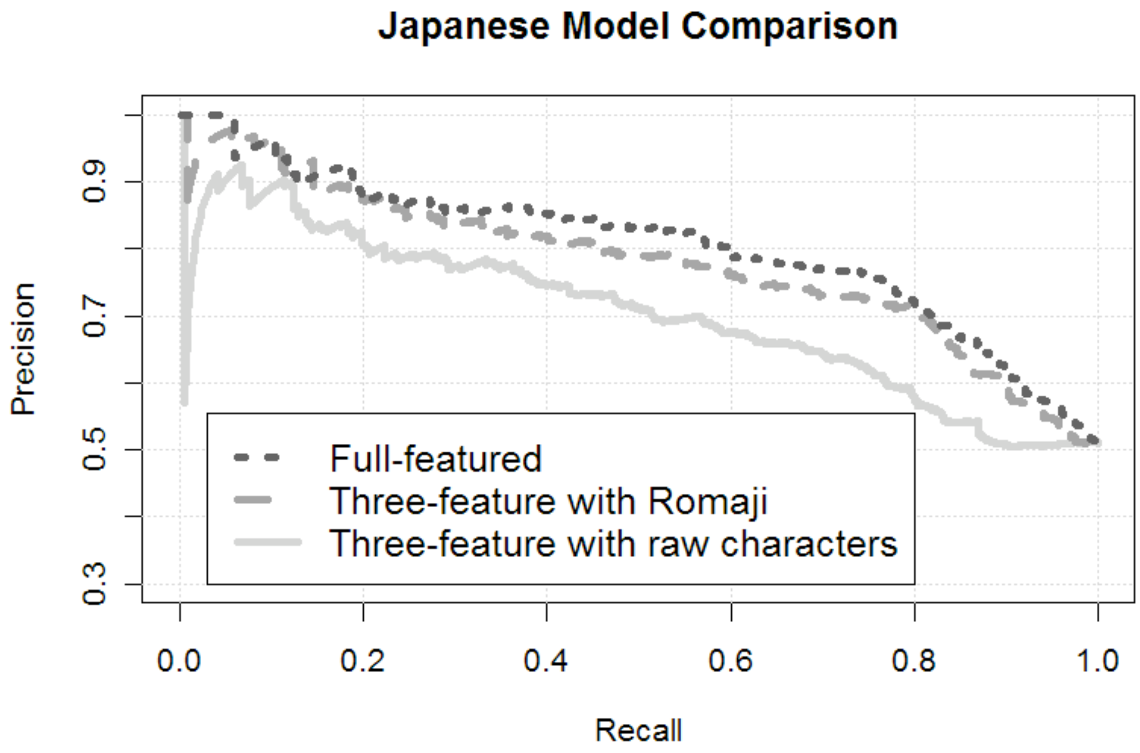

**Figure 1. Comparison of Japanese models before and after Romaji normalization.**

## 8 Homophone filtering and exception lists

Homophone rewrites are often good rewrites, as evidenced by the importance of normalized character edit distance in our learned ranking function for query rewrites. However, in some cases substituting a homophone leads to a bad rewrite. In this section we describe several of these cases, and give filtering rules based on them which further improve performance.

For most kanji compounds, homophones are different in meaning and it is fairly easy to distinguish between them. For example, 端, 橋, and 箸 all have the hiragana representation はし (*hashi*), yet represent *bridge*, *edge* and *chopsticks* respectively. An exception is the case of personal names, in which the kanji variation, mostly due to historic use, leads to several different ways to represent the same name: for example, 渡邉, 渡部and 渡辺 are all kanji representations of family name *Watanabe*. This may be because it is easier to confuse proper kanji in family names than in the words with very different semantics. Suggesting different variants for family names can therefore be seen as desirable, whereas in other cases would lead to significant topic shifts with respect to the original query.

To address the homophone problem, we filtered out all kanji-to-kanji whole query and phrase substitutables that have the same kana form and are not personal names. Although the number of such cases is relatively small – only about 3,000 phrase substitutables out of about 2 million – this does improves precision. Performance using this rule is shown in Table 11 as "kanji-kanji-filter." Katakana-to-kanji rewrites with romaji edit distance of zero also tended to have poorer quality than most pairs with zero edit distance. For instance, the company "alc" has the katakana representationアルク (aruku), which led our rewriting system to suggest, mistakenly, 歩く (aruku; to walk). We concluded that a katakana query almost always carries a meaning distinct from same-sounding kanji, and blocked katakana-to-kanji rewrites. Performance using this rule is shown in Table 11 as “katakana-kanji filter.” Finally, we excluded largely meaningless rewrites containing a query with only a single character. Applying this along with the other filters, we obtain the model we call “clean.” In the end, we got an average precision of 84% compared to the overall precision of 54%.

| Filter | None | Katakana-kanji filter | Kanji-kanji filter | Clean |
|---|---|---|---|---|
| Average Precision | 81.8 | 82.6 | 83.6 | 84.3 |

**Table 11 Average precision of automatic query rewrite quality with filters to remove rewrites of katakana to kanji, and kanji to kanji with the same pronounciation.**

## 9 Conclusion

Japanese query logs contain a mix of characters from the set of character sets used in Japan and internationally. We have empirically measured this mix, showing for example that surprisingly, Japanese queries typically contain at least one space, suggesting that web searchers have learned to modify their queries to improve search results. We have also shown that we searchers tend to modify the character sets they use in query sessions, since we can find similar queries by identifying those that differ by small amounts after normalizing across character sets. Query and phrase session substitutables are an effective means of identifying semantically related Japanese queries across a range of Japanese writing types. With upwards of 80% precision, we can identify equivalent substitutes for 43.6% of Japanese queries. Taking into account the Japanese writing system leads to significant improvements, both for features and model weights, and also through Japanese-language-specific homophone blocking rules.

## References

American National Standards Institute. (1972) ANSI Z39.11-1972 American National Standard System for the Romanization of Japanese. New York, American National Standards Institute.

Basis Technology (2006) BasisTech Knowledge Center. http://www.basistech.com/knowledge-center.

Chikamatsu N., Shoichi Y., Nozaki H., Long E. (2006) Development of Japanese Logographic Character Frequency Lists for Cognitive Science Research. http://nozaki-lab.ics.aichi-edu.ac.jp/nozaki/asahi/yes.html.

Halpern J. (2001) Outline of Japanese Writing System. Kanji Dictionary Publishing Society.

Halpern J. (2001) The Complexities of Japanese Homophones. The CJK Dictionary Institute.

Jones R., Fain D.C. (2003) Query word deletion prediction. SIGIR-2003, pages 435-436.

Jones R., Rey B., Madani O., Greiner W. (2006) Generating Query Substitutions. WWW2006, Edinburgh, UK.

Kapur, S and Parikh, S (2006) Unity: Relevance Feedback using User Query Logs. SIGIR 2006.

Makino H., Kizawa M. (1980) An Automatic Translation System of Non-Segmented Kana Sentences into Kanji-Kana Sentences. COLING80, pp. 295-302.

Manning C.D., Schütze, H. (1999) Foundations of Statistical Natural Language Processing. MIT Press.

Nagata M. (2000) Synchronous Morphological Analysis of Grapheme and Phoneme for Japanese OCR. Proceedings of ACL 2000, pp. 384-391.

Ruthven, I. (2003) Re-examining the potential effectiveness of interactive query expansion. SIGIR-2003.

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

Spink, A. and Jansen, J. (2004) Web Search: Public Searching of the Web (Springer Publishers )

Terra E., Clarke C. L. A. (2004) Scoring missing terms in information retrieval tasks. ACM CIKM-2004, pages 50-58.