A hybrid approach to fuzzy name search incorporating language-based and text-based principles

Paul Wu Horng-Jyh; Na Jin-Cheon; Christopher Khoo Soo-Guan

Nanyang Technological University, 31 Nanyang Link, Singapore 637718

Correspondence to: Paul Wu Horng-Jyh, Nanyang Technological University, 31 Nanyang Link, Singapore 637718. E-mail: hjwu@ntu.edu.sg

Abstract.

Name Search is an important search function in various types of information retrieval systems, such as online library catalogs and electronic yellow pages. It is also difficult due to the high degree of fuzziness required in matching name variants. Previous approaches to name search systems use ad hoc combinations of search heuristics. This paper first discusses two approaches to name modeling—the natural language processing (NLP) and the information retrieval (IR) models—and proposes a hybrid approach. The approach demonstrates a critical combination of complementary NLP and IR features that produces more effective fuzzy name matching. Two principles, *position-as-attribute* and *position-transition-likelihood*, are introduced as the principles for integrating the advantageous aspects of both approaches. They have been implemented in an NLP- and IR- hybrid model system called Friendly Name Search (FNS) for real world applications in multilingual directory searches on the Singapore Yellow pages website.

Keywords: Fuzzy Name Search; Natural Language Processing; Information Retrieval; Hybrid System; Language and Text

JIS Editorial Process
Received: 23rd August 2005
Revised: 8th March 2006
JIS-0238 v1.0

1. Introduction

Name Search has been an important search function in digital library and various types of information retrieval systems, such as online library catalogs, bibliographic retrieval systems, and electronic yellow pages. Recent research has suggested that it is prevalent in the World Wide Web as well [1]. A person's name can exhibit many variations in published documents, and users searching for a name may enter a variant form not found in the documents and text, or not matching the form indexed in the system. For example, a user issuing an author name search as "Lee Kuanyew" is likely to miss a record indexed as "Lee, Harry Kuan Yew″ although both refer to the same person–the first Prime Minister of Singapore. Similar instances involving slight mis-spelling or mis-ordering in the queries will result in fruitless name searches.

Yet few systems offer fuzzy name matching to help users retrieve records with variant person names. Name searches fail not only because of errors in the users' search query, but also because names have numerous acceptable variants. It is not a straightforward task to overcome the ambiguity due to the quantity of name variants. For instance, the name "Harry Kuan Yew Lee" is a Chinese name with native and adopted name tokens, where "Lee" is the surname, "Kuan Yew" the given name, and "Harry" the adopted name. Similarly, "Nurdini Abu Bakar Aljunied" is a Malay name with an adopted (Arabic) surname "Aljunied." To illustrate this, different forms of a name are listed in Table 1 according to five types of variations: Name Alternation (NMA), Sound-alike (SAL), Abbreviation (ABB), Short-hand (SHH), and Contraction (CON). To further quantify the fuzziness of the name, the estimated number of variants is also included in the table.

Therefore, an automatic name search system will need to incorporate fuzzy name searching capability that can retrieve the target name despite the multitude of legitimate name variants. Many ad hoc techniques have been applied in search systems that attempt to handle the complexity demonstrated in Table 1, for example, [2] and [3]. There are also Natural Language Processing (NLP) and Information Retrieval (IR) techniques that have been applied. They focus on certain aspects of the issues, such as the sound aspects [4] and the token position aspects (e.g., [5]). Fundamentally, they are motivated by distinct theories of the nature of names, namely:

JIS Accepted for Publication
By the Journal of Information Science: http://jis.sagepub.co.uk

Table 1. Possible Variations of Person Names

Type of Variation	Name Instances	Estimated number of variants
Name Alternation (NMA)	*Harry Kuan Yew Lee* or *Kuan Yew Harry Lee*	$(N-1)!$ where N is the number of name tokens (=6, when N=4)
	Nurdini Abu Baker Aljunied or *Aljunied, Nurdini Abu Baker*	
Sound-alike (SAL)	*Harry Kuon You Lee* or *Harrie Kuan Yew Lee*	$\prod_{i=1}^{i=N} \lvert W_i \rvert \cdot (N-1)!$ where $\lvert W_i \rvert$ is the number of characters in the name token W_i ($=\sim 4^4 \times 6 = 1536$ assuming N=4 and $\lvert W_i \rvert = 4$)
	Noordini Abu Baker Aljunie or *Nurdini Abu Baker Aljuneid*	
Abbreviation (ABB)	*Harry K Y Lee* or *H. Kuan Yew Lee*	$(2^{N-1}-1) \cdot (N-1)!$ (= 42, when N = 4)
	Nurdini A. B. Aljunied or *Nurdini Abu Baker A.*	
Short-hand (SHH)	*Hari Kuan Yew Lee* or *Har Kuang Yew Lee*	Similar to SAL (=~1536 assuming $\lvert W_i \rvert = 4$)
	Dini Abu Barker Aljunied or *Nur Abu Baker Aljunied*	
Contraction (CON)	*Harry Kuanyew Lee*	Similar to NMA (=6 when N=4)
	Nurdini Abubaker Aljunied	

Name-As-Language (NAL): Person names follow a conventional style of writing – a kind of grammar. More specifically, names can be parsed into components such as surname, given names and other limited number of name attributes.

Name-As-Text (NAT): Person names are a text consisting of tokens as indexing features. More specifically, methods such as relevancy ranking and query expansion can be applied to rank name search results and accommodate name variants.

Previous name search systems are reviewed in Section 2. In Section 3, we discuss the Name-As-Language view through a review of NLP techniques applicable to automatic name search systems. The principle of *position-as-attribute (PAA)* is proposed to reflect the Name-As-Language perspective of fuzzy name matching. In Section 4, the Name-As-Text view and IR techniques are discussed, and the principle of *position-transition-likelihood*

(PTL) is proposed to reflect the Name-As-Text perspective. A hybrid model of an automatic name search system called Friendly Name Search is introduced in Section 5 that accomplishes the critical combination of the two principles.

2. Previous and related work on Name Search

Systems which deal directly with the task of automatic name search include the *Synoname* system [3, 6], developed by a team under the Getty Art History Information program to archive art works by around 6,000 artists. When museums exchange cataloging information, without a proper name matching procedure, artworks by the same artist may be cataloged under different names. The system's engine for name matching includes 12 comparison techniques: (1) Exact match, (2) Omission of one character, (3) Substitution, (4) Transposition, (5) Difference in punctuation, (6) Initials, (7) Extended name, (8) Inclusion of names within names, (9) No first name, (10) Word approximation, (11) Confusion of dividing names, and (12) Character approximation.

The first 4 techniques concern fuzzy string matching within 1-Levenshtein distance [7]. Techniques (7), (8), (9) and (11) are easily handled by an IR system, since they just mean the set of features in the query string is a subset of those of the name record. Technique (10) presents a challenge which can be covered under our considerations of Sound-alike (SAL) variants. Technique (6) is exactly the same as those treated in Abbreviation-type (ABB) variants. This leaves only technique (12), which is character approximation. For example, "Backhuyzen, Ludolf" is related to "Bakhuysen, Ludolf," and "D'Espagnat, George" to "Espagnat, George d'". These examples show that even strings with two Levenshtein distances away still need to be regarded as a match. Thus, technique (12) is just a generalization of techniques which handle cases of only 1-Levenshtein distance. In general, with the capability for handling NMA-, SAL-, ABB-, SHH-, and CON-types of name variation, our approach can have the same flexibility in name matching as Synoname.

Another interesting study on Name Search is a Ph.D. research by Hermansen [2], who investigated the "New York State Identification and Intelligence System." The two important aspects of names, form and sound (identified as "name structure" and "transliteration" in the thesis), were argued to be crucial for automatic name search systems. However, no system implementation was involved in the study. Also, as it was a rather early work, no reference to modern NLP or IR techniques were made. On the other hand, many aspects of the ad hoc name search algorithms were examined, providing a good review for the technologies available up to the mid 80's. These ad hoc techniques are: sound-based similarity [8, 9], n-gram entropy [10], name subsetting [11], and record linkage [12]. Pfeifer, et al. performed experiments for measuring retrieval effectiveness of various proper name search methods [13]. They argue that phonetic similarity (PHONDEX) works as well as typing errors (Damerau-Levenstein metric) and plain string similarity (*n*-grams), and the combinations of these different techniques perform much better than the use of a single technique. Thompson and Dozier demonstrates the

prevalence of names in the news and legal domains and quasi-name search mechanisms, such as proximity operators applied in the query string, can be applied to increase information retrieval effectiveness [14]. Navarro, et al. applied fuzzy string matching algorithms and mechanisms to the retrieval of proper names, given emphasis to those that are of Portuguese and Spanish origins due to the geographical and culture convection of the application: Latin American [15]. Although their achievement is impressive, it is based on a different theory – stringology, to natural language processing and information retrieval of concern in the paper; this is appropriate given its specific geographical and culture considerations.

Borgman and Siegfried highlighted two related application areas of fuzzy name matching: name authority control and duplicate record detection [6]. More recently, new application areas have been explored in text mining and machine translation. Beli and Sethi discussed potential matching algorithms for patient identification resolution for use with a massively distributed Master Patient Index, which is a facility to make all patients' medical records in the U.S. accessible to care providers [16]. The patient identification resolution relies on additional attributes in addition to name, such as address, telephone, social security number, and date of birth.

Name searching is also important in the fields of machine translation and cross-lingual information retrieval. Stalls and Knight [17] and Virga and Khudanpur [18] worked on translating names and technical terms using phonetic translation (e.g., from English to Mandarin). Pirkola et al. [19] investigated a fuzzy cross-lingual translation of proper names and technical terms, but no phonetic elements were included in the techniques. Brill, et al. [20] proposed an automatic harvesting mechanism for Katakana-English term equivalence pairs. Both Pirkola and Brill's work can be used in cross-language information retrieval. In fact, Larkey, et al. [21] explored static and transliterated English-to-Arabic proper name translation mechanisms in cross-language information retrieval. It was found that proper name translation improves the retrieval effectiveness. However, as multiple forms of the name translation exist, fuzzy name searching remains indispensable when non-canonical name forms occur. Thus, it seems most of the research on applying machine translation to cross-language information retrieval is yet to be subject to due consideration and points raised in this paper would be beneficial for them to incorporate.

Lastly, in the emerging field of text mining, Patman and Thompson [22] demonstrated that the capability to process names is crucial in named entity recognition, and in entity matching in co-reference resolution, on which the success of text mining rests. Many of the issues discussed in this paper will be relevant as it provides a sound framework for designing effective fuzzy name matching mechanism that is necessary in text mining.

In summary, existing name searching systems use mostly ad hoc techniques originating from disparate fields of study, such as fuzzy string matches, rule-based pattern matching, record-linking, and soundex schemes. These components optimize particular aspects in fuzzy name searching, possibly introducing problems in other aspects of name searching. In contrast, our system, the Friendly Name Search system, adopts a combination of two approaches which addresses the balance between two competing theories for automatic name search system

development. However, it is acknowledged that the form and sound aspects of names across the world are largely still an ad hoc phenomenon. Thus, in an operational environment, ad hoc methods may still be required to address certain peculiarities.

3. **Name**-As-Language -- The Deep Structure of Names

Similar to the generative grammar perspective in linguistics, the Name-As-Language theory assumes that names, like sentences and other text units, can be generated in a rule-governed fashion by a name grammar. A relevant question is:

> What remains *constant* throughout its conventional variants? That is, what is the "deep structure" of names?

There are two aspects to the deep structure of names: the form and the sound of names. To address fuzzy name matching, technologists with the Name-As-Language view would build name search systems based on the *normalization* of names aiming at recovering the deep structure that is constant across its variants. After normalization, the search process becomes a relatively straightforward template matching of attributes and values.

With regards to the form of names, as demonstrated in Fig. 1, a candidate architecture based on the Name-As-Language view may consist of several processing steps. First the surface forms of the name, "*Harry Lee Kuan Yew*" and "*Kuan Yew Lee Harry,"* are processed by a module called Name Classifier, which determines the name type of the names. In this case, it is determined to be of CH (Chinese) name type. Second, the regular rule set for CH name type is activated. Together with input from a Name Attribute Dictionary, a module called Name Parser runs through a regular grammar parsing process based on the activated rule set, disambiguating the name attributes and producing the same normalized name template – "*SN: Lee; GN: Kuan; GN: Yew; AN: Harry.*" *SN* stands for Surname, *GN* for Given Name, and *AN* for Adopted Name. In sum, normalization based on Name-As-Language viewpoints can be achieved by a system, which consists of the following normalization components: (1) Name Classifier; (2) Name Attributes Dictionary; (3) Name Grammar – Regular Rule Sets (4) Name Parser. The system recognizes legitimate name variants for each name type and transforms them into a normalized form despite the variation in the surface forms.

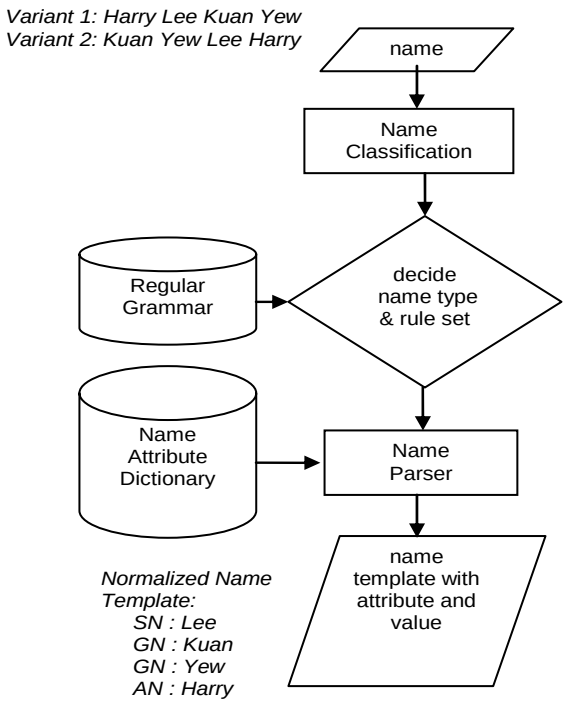

Fig. 1: A candiate normalization architecture based on the Name-As-Language viewpoint

More results of the above normalization process are demonstrated in Table 2, including recognition of names such as, "Harry Kuan Yew Lee" (Chinese), "James Hla Gyaw" (Burmese), and "Savar Sankaran Narashimhalu" (Indian).

Table 2. Attribute-Value Templates for Chinese, Burmese and Indian Names

Name Attributes	*Harry Lee Kuan Yew*	*James Hla Gyaw*	*Savar Sankaran Narasimhalu*
Surname (SN)	*Lee*	-	-
Native Given Names (GN)	*Kuan Yew*	*Hla Gyaw*	*Narasimhalu*
Adopted Name (AN)	*Harry*	*James*	-
Father's Name (FN)	-	-	*Sankaran*
Place Name (PN)	-	-	*Savar*

Table 3 and Table 4 illustrate the kinds of entries found in the Name Attribute Dictionary and the Regular Grammar Rule Sets, which are required in the normalization process.

Table 3. Sample Entries of a Dictionary for Chinese (CH), Malay (MY), and Indian (IN) Names

Name Type	Name Token	Name Attribute	Name Type	Name Token	Name Attribute	Name Type	Name Token	Name Attribute
CH	Lee	*SN;* *GN*	IN	Vimol	*SN* *FN*	MY	Husain	*GN* *FN*
CH	Kuan	*GN*	IN	Goel	*SN*	MY	bin	*SON-OF*
CH	Yew	*SN* *GN*	IN	Pillai	*GN*	MY	Umar	*GN* *FN*
CH	Harry	*AN*	IN	Nair	*CN*	MY	Sharifah	*NN*
CH	Annie	*AN*	IN	Gurmi	*GN*	MY	Aishah	*GN*
CH	Tan	*GN*	IN	Singh	*CRT*	MY	Ahmad	*GN*
CH	Seow	*SN* *GN*	IN	Sethi	*Title*	MY	Alsagah	*SN*
CH	Ah	*GN*	IN	Abdur	*SO*	MY	Datok	*Title*

Name attribute acronyms*: AN:* Adopted Name; *CRT:*Clan-Religion Title; *CN*: Clan Name; *FN*: Father's Name; *GN*: Given Name; *NN*: Nobility Name; *SN*: Surname; *SO:Slave-Of.*

Table 4. A Sample Grammar for Chinese (CH), Malay (MY), and Indian Name Templates

Name Type	Regular Grammar Rules	Examples
CH Type 1	SN [-SN] GN [GN] [,[AN]+]	*Lee Kuan Yew, Harry*
CH Type 2	([AN+] | GN [GN])+ SN [-SN]	*Harry Kuan Yew Lee*
CH Type 3	Type 1 NEE Type 1	*Tan Annie NEE Goh Ah Muei*
CH Type 4	Title [Type 1 | Type 2]	*Mr. Harry Kuan Yew Lee*
IN Type 1	GN [FN] SN	*Vimol Goel* (Hindi names)
IN Type 2	[FN | PN] + GN	*Savar Sankaran Narasimhalu* (e.g., Tamil)
IN Type 3	GN CRT [Title]	*Gurminda Singh Sethi (e.g., Punjabi)*
IN Type 4	GN [CN] [HT]	*Abdur Rahim Khan Syed (e.g., Urdu)*
MY Type 1	GN+ [bin | binte] FN+	*Husain bin Umar*
MY Type 2	NN Type 1 [SN]	*Sharifah Aishah Ahmad Alsagaf*
MY Type 3	Title Type 1	*Datok Hussein Onn*

The above illustration based on sample linguistic knowledge may seem straightforward. However, as will be explained in the next section, the recognition of the deep structure of a name is a difficult task. It requires sorting through the complexity of name variants and transforming them into the normalized form. Only knowledge as complex and comprehensive as a native-speaker's language intuition will be able to achieve an effective level of recognition.

Next, with regards to the sound of names, the deep structure of a name token is simply its phonemes. However, name transliteration increases the complexity of the problem. For example, there is likely to be only one Arabic spelling in Arabic script for a name like "Sulayman." However, there are many Sound-alike (Spell-alike) versions in romanized forms such as "Suliman", "Seleiman", and "Solomon". The same phenomenon is observed in Chinese names where "Zhi," in standard Hanyu Pinyin, can be spelt as "Jih", "Jyh", "Ji", "Chi", "Chih" and so on. This is demonstrated in Table 5[1].

Table 5. Arabic and Chinese Name Structure (Sound)

Surface Names	Phonetic Transcription
Sulayman, Suleiman	s.u.l.ey.m.ax.n
Salayman, Seleiman, Sylayman	s.ax.l.ey.m.ax.n
Suliman	s.u.l.ih.m.ax.n
Solomon	s.ao.l.ao.m.ao.n
Zhi, Jih, Jyh, Ji	zh.i
Chih, Chi	ch.i

3.1 Challenges to Name Search Systems Based on the Name-As-Language Viewpoint

The grammatical formulation of person names has the advantage of being precise. The disadvantage, as mentioned, is that it needs to be exhaustive. To achieve the degree of human language intuition by encoding the knowledge exhaustively is very time consuming. One such example is the Anapron system developed by Golding [4] to pronounce names of different ethnic origins. The system contains 90 language identification rules, 205 morpheme rules, 619 transcription rules, and several hundred rules on syllable and stress structure assignments. Even with all this effort, the system only covers the sound variants. Many more rules will be needed to cover the form variants shown in Table 1.

1 The transcription is based on the DECtalk system.

An alternative to the deterministic grammar-rule-based approach is the data-oriented empirical approaches. These approaches trade the manual rule encoding and dictionary building efforts with those of data collection and manual annotation, followed by statistical training and optimization. For instance, a similar system to Anapron, which normalizes sound variants, is proposed in [23]. It uses a Hidden Markov Model (HMM)-based name classifier. Its Spanish name classifier is a HMM consisting of 8 states and is trainable by sample data of Spanish names. Similarly, Bosch and Daelemans [24] described a data-oriented method for grapheme-to-phoneme conversion, whereby a statistical measure, *Information Gain,* is applied to induce rules for transcribing Dutch words.

An empirical approach to normalize form variants of names has not been attempted previously. However, the task of assigning name attributes to name tokens can be seen as similar to that of assigning grammatical tags to words. Church [25] and DeRose [26] applied a Hidden Markov Model (HMM) to tag English sentences with grammatical categories. In fact, one such real world example for normalizing other types of phrasal units, such as addresses, has been developed using the same approach as grammatical tagging [27]. It would be feasible to apply the same mechanism to tag name with name attributes.

In general, the empirical approach requires a tagged corpus to develop the model. Such a corpus seem easier to construct compared to dictionaries and grammatical rules for names. Another advantage is that this approach is nondeterministic, i.e. more than one plausible processing result can be computed, which allows further spelling disambiguation to be applied in the case of uncertainty.

3.2 Position-As-Attribute (PAA) Principle

Whether it is a rule-based or an empirical approach, a name search system based on the Name-As-Language viewpoint requires much manual effort and time, to build the dictionary, thesaurus, schema/grammar rule, and linguistic tagging resources. This will most likely render a Name-As-Language approach infeasible, especially if one considers the need to assign attributes, such as Surnames, Given Names, Family Names, Castle Names, Titles, etc., to name tokens in all the culture and language groups in the world.

We propose a resource economical approach called the *position-as-attribute (PAA)* principle, which is simply:

Name positions are literally taken as attributes for the name template

Table 6 illustrates the result of applying this principle to the example names used earlier in Table 2. With positions as the attributes/tags and using a dynamic programming-like constraint checking and scoring process, the most plausible positions for each name token is identified and used as the basis for ranking and retrieving actual name records. More details are given in Section 5 of the paper.

Table 6. Position-As-Attribute Applied to Example Chinese, Burmese and Indian Names

Name Attribute	*Harry Lee Kuan Yew*	*James Hla Gyaw*	*Savar Sankaran Narasimhalu*
Position1	Harry	James	Savar
Position2	Lee	Hla	Sankaran
Position3	Kuan	Gyaw	Narasimhalu
Position4	Yew	-	

Since the positional information of a name token can be readily accessed from a name database, no additional manual tagging is necessary during this process. That the *position-as-attribute* principle is plausible can be seen from the fact that in a fully normalized name database, controlled by cataloguing rules, the position corresponds exactly to the attribute of the name tokens. For instance, the first token for both Chinese and English names is the surname.

Taking the surface structure literally as the deep structure reduces resource overhead. This approach can also be applied to the sound aspect of names. In fact, Bosch and Daelemans [24] used a straightforward table look-up method to transcribe the sounds of a Dutch name, outperforming statistical approaches such as those based on Information Gain. Here lies also one intrinsic difficulty of the Name-As-Language view: names, like most languages, have idiosyncrasies that defy any systemic modelling based on grammar rules. In fact, as will be discussed in the next section on the Name-As-Text view, it is more effective to deal with these idiosyncrasies within the standard treatment of information retrieval. Conversely, there are certain problems in the Name-As-Text approach that are better handled in the Name-As-Language approach. A balance between the two views provides a more promising solution for the complexity posed by fuzzy name search system.

In summary, what are gained by examining the Name-As-Language view of name search systems and the *position-as-attribute* principle are as follows:

1. Recovering the deep structures of the form and sound aspects of names makes name search systems more effective. But the resources required by the normalization process can be a serious constraint.
2. To overcome resource overhead, it is necessary to take readily available information from the name data as the basis for language modelling. The position-as-attribute principle is proposed.

4. Name-As-Text -- The Feature and Similarity of Names

From a Name-As-Text perspective, names are seen as sets of characteristic features consisting of name tokens. For example, in a name database consisting of "Harry Kuan Yew Lee", "Harry Hui Kuan Deng", "Jack Yew Hui

Lee" and "Peter Wu," nine distinct name tokens are identified: (1) "Harry," (2) "Kuan," (3) "Lee," (4) "Hui," (5) "Deng," (6) "Jack," (7) "Yew," (8) "Peter," (9) "Wu." Each name is then represented by a feature set, or feature vector, as shown in Table 7. Standard information retrieval techniques are used in the name search.

Table 7. Name Records and Feature Representations

Name Records	Feature Representation
Harry Kuan Yew Lee	{1, 2, 7, 3}
Harry Hui Kuan Deng	{1, 4, 2, 5}
Jack Yew Hui Lee	{6, 7, 4, 3}
Peter Wu	{8, 9}

One of the fundamental assumptions of the feature representation in the Name-As-Text view (as exemplified in many information retrieval models) is that each of the name tokens is represented by a literal string. Thus, the retrieval of the name requires the query name tokens to match exactly with the corresponding feature tokens that were indexed in the first place. A query such as "Kuan Yew Lee Harrie" is first translated into a query feature {2, 7, 3, 10}, where (10) represents "Harrie". It cannot retrieve any of the records listed in Table 6, as a result of requiring exact string match, although it was meant to retrieve "Harry Kuan Yew Lee."

Query expansion is the standard method used in information retrieval to increase recall (i.e. increase the number of relevant records retrieved). Searching for "Sound-alike" name variants, such as "Harrie" and "Harry", can be treated as a "Meaning-alike" expansion process, in which query terms are expanded with synonymous terms using a thesaurus [28, 29]. Traditionally, a thesaurus, or a Name Equivalence Database in our case, is constructed manually. IR researchers have explored automatic methods of thesaurus construction through statistical means, such as mutual information [30] or clustering algorithm [31, 32]. Fig. 2 illustrates a system architecture for query expansion.

Query expansion increases retrieval recall such that relevant records are retrieved despite "under-specification" in the query. However, it has a potential negative side-effect--the precision rate may decrease inadvertently. Most of the empirical findings point to the inverse relationship between Recall and Precision: improvement in either tends to be associated with poorer performance of the other [33]. The balance can be achieved by implementing additional measures to maintain the precision rate in spite of the increased retrieval results. Thus, the challenge becomes how to introduce a scoring mechanism for retrieved names that assign higher rankings to relevant results. This is discussed in the next section.

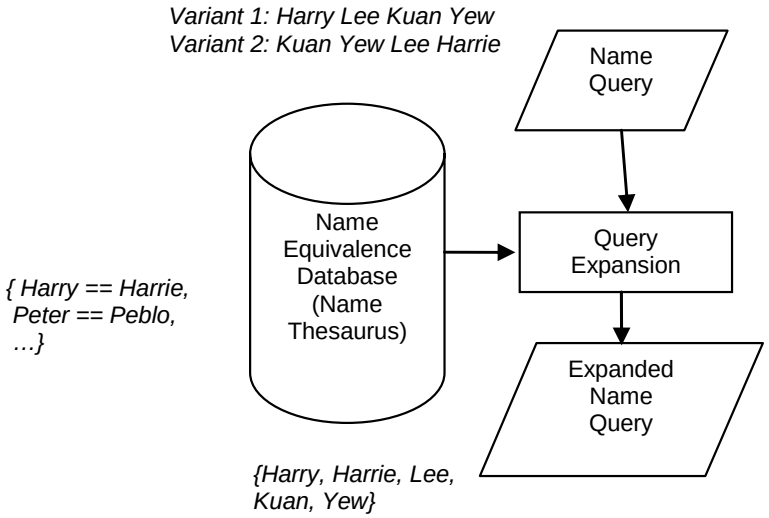

Fig 2. A query expansion architecture based on the Name-As-Text view

4.1. Challenges to Name Search Systems Based on the Name-As-Text Viewpoint

Most IR approaches rank search results based on a similarity scoring formula via a term weighting approach, and employ relevance feedback to improve the term weighting [34, 35]. A variety of term weighting schemes based on term frequency *(tf*) and inverse-document frequency *(IDF)* may be used. The basic idea is that the less frequently a particular token *i* appears across the collection (i.e. the lower the document frequency, and the higher the inverse document frequency, IDF_i) the more characteristic the token is to the name record. Similarly, the more frequent a token occurs in a name record (term frequency), the more characteristic the token is to the name. However, *IDF* and *tf* overlook the name attribute aspect of the token treated in the Name-As-Language view. For instance, a database may contain target records (A) "Robert Kong Kong Tan," where "Tan" is the surname, and (B) "Robert Kong Tan," where "Kong" is the surname. For a query "Robert Kong," record (A) would be retrieved with a higher relevancy score than record (B) due to the higher frequency (and consequently higher weight) of "Kong" in record (A). A name attribute analysis would indicate that record (B) is a better match because "Kong" in "Robert Kong" is more likely to be a surname – an interpretation more consistent with record (B). The issue of name attribute will be discussed below in the context of the "dimension" of query expansion.

For name records, unlike free-text records, the order of the name tokens is significant for distinguishing names. A name can have more than one correct order, as illustrated earlier in Table 1. For instance, a name like *Harry Lee Kuan Yew*, represented as the sequence <1, 3, 2, 7>, is the same as *Harry Kuan Yew Lee* (<1, 2, 7, 3>), while *Harry Yew Kuan Lee* (<1, 7, 2, 3>) is a different name. The feature representation of a set of name tokens will not

be able to distinguish the difference as they are both represented as {1, 2, 7, 3}. Thus, to measure the similarity between different name orderings precisely despite having expanded query terms is a central consideration in the scoring mechanism.

Some IR approaches do take into account the position and order of tokens. Experiments on ranked retrieval output have revealed that token position can improve retrieval precision. Among several scoring schemes based on position, sentence boundary, and order of position, proximate matching token pairs in sentences have been found to be the most effective [5, 36, 37]. Similarly, *phrase* and *proximity* operators, such as adjacency, window size and directed window, are adopted in traditional IR systems [38, 39]. However, little consideration has been given to these scoring schemes in the context of Query Expansion as a Precision-enhancing device that counter-balances increase in Recall. On the other hand, certain "Meaning-alike" expansion approaches with automatic relevance feedback have proven to increase the retrieval effectiveness. Upon closer examination, counter-balance measures have been applied in the automatic relevance feedback process by maintaining the original query term and its expanded terms in the same cluster or dimension, and using these dimensions as constraints in the retrieval [40]. A similar concept of independence was used by [41]: in an attempt to improve the precision of short search queries, the results were filtered by a Boolean formula in conjunctive normal form where each conjunct is intended to correspond to an independent subtopic of the query. For “Name-equivalence” expansion in a name search system based on the Name-As-Text view, similar measures of “dimensionality” in the expanded tokens need to be incorporated. As mentioned earlier, the name attributes, such as surname and given name, are the natural search dimensions. In the following, two kinds of constraints are introduced to explain the concept of dimensionality further.

There are essentially two approaches to ranking the relevance of records retrieved:

- the expanded tokens can be combined with the original tokens, upon which the accumulation of postings is done for each token in the combined query. The expanded tokens are not distinguished from the original tokens. We refer to this as the +-combination.
- postings in each of the *Cartesian* products of the expanded tokens can be accumulated. We refer to this as the ×-combination

As mentioned, earlier “meaning-alike” query expansion techniques adopt the +-combination type (e.g. [28]). However, in the case of "Name-equivalence" type of name variation, the distinction in dimensionality cannot be maintained by just a simple +-combination.

Take the example query string “Kon Yang Kong” targeting to retrieve the name “Kon Yang Khon”. The terms in the query string can be expanded as follows:

- "Kon": {"Kon", "Kong", "Khon"}
- "Yang": {"Yan", "Yen", "Yang"}
- "Kong": {"Kon", "Kong", "Khon"}

The +-combination query expansion would yield {"Kon", "Kong", "Khon", "Yan", "Yen", "Yang", "Kon", "Kong", "Khon"}, where positional information is not maintained. Thus, a name such as "Kon/SN Yang/GN Chee/GN"[2] (not the targeted name) would be retrieved with an equal relevancy score as "Kon/SN Yang/GN Khon/GN" (the targeted name), both getting 4/9 from Dice's coefficient[3]. This is because the dimensionality of "Kon", a surname, and "Khon", a given name, are collapsed into an indistinguishable group in the +-combination process.

In order to avoid the drawbacks described above, the ×-combination of query expansion has to be adopted. However, computationally, the ×-combination may lead to combinatorial explosion. For example, if a name has m tokens $<A_1,\ldots,A_m>$, and each yields $|A_i|$ tokens after query expansion, the commonly used +-combination will result in $\sum_{i=1}^{i=m}|A_i|$ query expanded tokens, whereas the ×-combination will yield $\prod_{i=1}^{i=m}|A_i|$ tokens. How to overcome this serious constraint is the topic of the next section.

4.2. Position-Transition-Likelihood (PTL) Principle

To restrict the combinatorial expansion in the ×-combination query expansion, one approach is to incorporate a filtering mechanism while maintaining the same dimensionality. We propose the *position-transition-likelihood* principle:

> The likelihood of a transition between a pair of name tokens, in terms of their positions, is used to filter the expanded queries in ×-combination.

To apply this principle to the example query string "Kon Yang Kong" and the expansion group for each term given earlier, the likelihood of the following nine transitions for *Kon->Yang* (and its expanded terms) for position 1 to 2 need to be calculated:

Kon_{pos1}->Yan_{pos2}, Kon_{pos1}->Yen_{pos2}, Kon_{pos1}->$Yang_{pos2}$,

$Kong_{pos1}$->Yan_{pos2}, $Kong_{pos1}$->Yen_{pos2}, $Kong_{pos1}$->$Yang_{pos2}$,

$Khon_{pos1}$->Yan_{pos2}, $Khon_{pos1}$->Yen_{pos2}, $Khon_{pos1}$->$Yang_{pos2}$.

Transitions for *Kon->Yang* for position 1 to 3, position 2 to 3, position 2 to 1, position 3 to 2 and position 3 to 1 (generally, position *n* to *m*, for *n*,*m*=1 to 3 and $n \neq m$) need to be calculated.

---

2 As shown in Section 2, SN stands for Surname and GN, Given Name.

3 Dice's coefficient is defined as $2(|Q \cap D|)/(|Q| + |D|)$

Out of all the possible ×-combinations of the expanded "Kon Yang Kong," only one (the correct one) was found to be plausible—"Kon Yan Khon," whose position transitions, Kon_{pos1}->Yan_{pos2} and Yan_{pos2}->$Khon_{pos3}$, had the highest likelihood score. The transition likelihood is derived from a database of names. More details about the likelihood scoring mechanism are discussed in Formula (1) in Section 5 below.

In summary, our analysis of the Name-As-Text view results in the following:

1. Traditional measures of similarity and relevancy in IR are not sufficient for automatic name search systems, although the lead time to an operational Name-As-Text-based system can be shorter than a Name-As-Language-based one. Positional constraints that were introduced based on the *position-as-attribute* principle need to be applied to make the search more precise.
2. Overcoming the combinatorial explosion for more precise retrieval is crucial. As such, the *position-transition-likelihood* principle is proposed for filtering out unlikely combinations generated by query expansion mechanism required for fuzzy matching.

5. Friendly Name Search (FNS) - Towards a hybrid theory-driven automatic name search system

A name search system called Friendly Name Search (FNS), aspects of which are patented[4], was developed incorporating the *position-as-attribute* and *position-transition-likelihood* principles [42]. The architecture of FNS system is shown in Fig. 3. The database name goes through a tokenization process before being indexed. During the process, domain specific name equivalence and metadata are incorporated in the name modeller to produce information based on the *position-as-attribute* and *position-transition-likelihood* principles. A Fuzzy Name Index is produced at the end of the indexing process. The query names are transformed and processed similarly as the names in the database, which are then matched, scored and ranked based on the fuzzy name indices to produce the search results.

4 "A System of Organizing Catalog Data for Searching and Retrieval" Patent No: US 6,381,607 B1 on 30 Apr 2002.

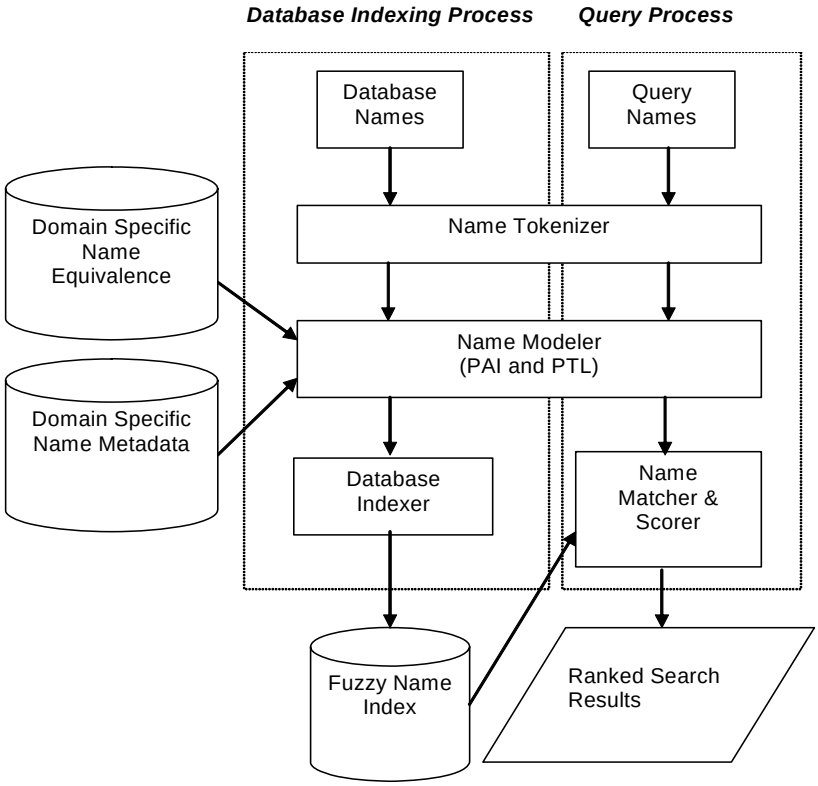

Fig. 3. Flow Diagram of Friendly Name Search

A version of the Friendly Name Search has been deployed in a Website, called *Singapore Yellowpages,* located at http://www.yellowpages.com.sg. The site contains millions of records and is among the most accessed Website in Singapore. Fig. 4 shows the result screen for the fuzzy query string "Kho Soo Gun". There are 15 matches, with "Khoo Guan Soon" ranked 1st, and "Khoo Soo Guan Christopher" ranked 4th.

The process by which FNS computes the results based on the *position-as-attribute* and *position-transition-likelihood* principles is demonstrated in Fig. 5. First, "Kho", "Soo" and "Gun", are expanded in the same manner as described in sections 4.1 and 4.2 above. The first token "Kho" is expanded into 3 tokens, "Kho", "Koh," and "Khoo." Similarly the second token "Soo" is expanded to "Soo" and "Soon", and the third token "Gun" expanded to "Gun", "Guan" and "Wang." Each expanded token has frequency counts for different positions. For example, the first expanded token "Kho" has frequency counts represented as Kho_{pos1} and Kho_{pos2}, for positions 1 and 2, while "Koh" has Koh_{pos1}, Koh_{pos2} and Koh_{pos3}, for positions 1, 2 and 3; similarly for other tokens, such as $Soon_{pos2}$ and $Wang_{pos3}$.

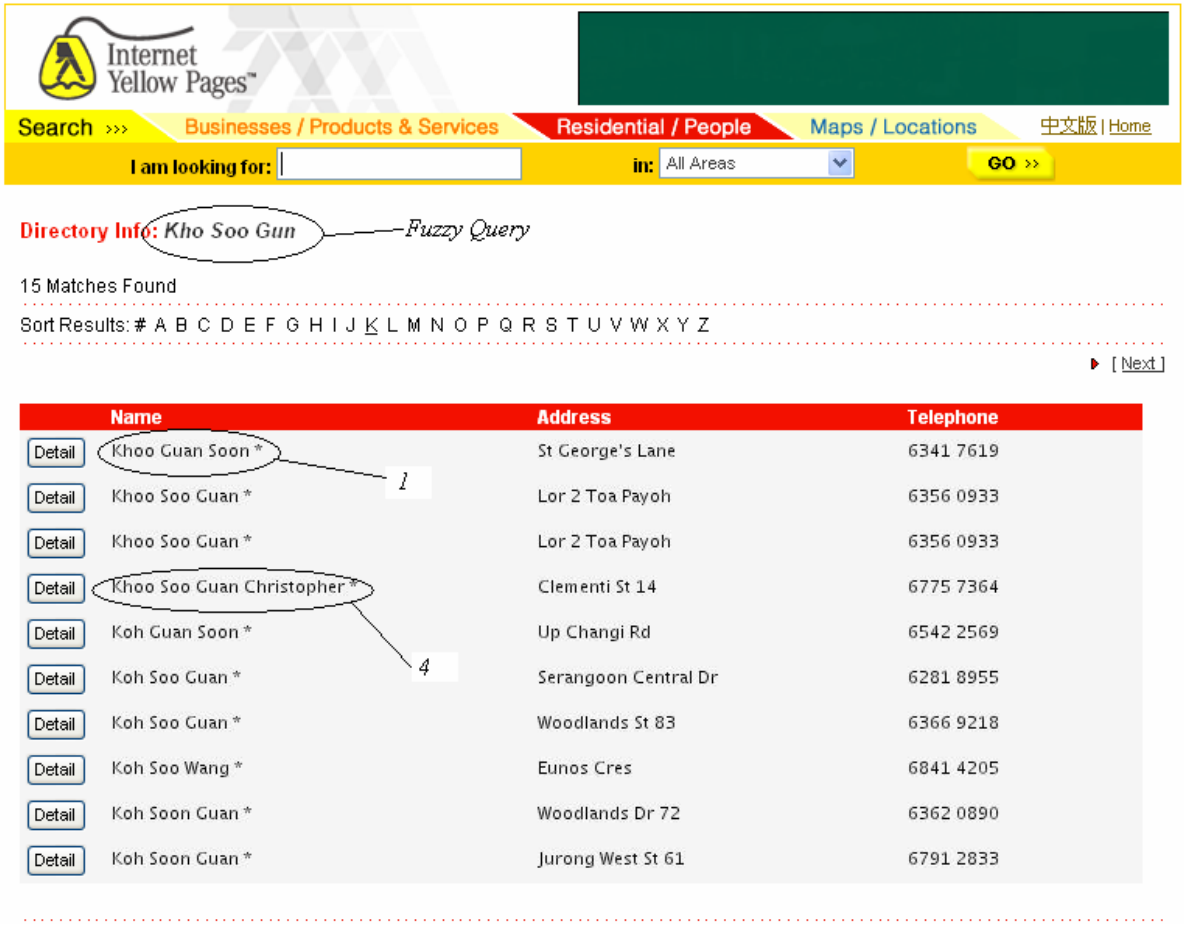

Fig 4. A Fuzzy Search Result on "Kho Soo Gun" in www.yellowpages.com.sg, where the directory search engine is powered by FNS

In Fig. 5, the tokens and their frequency information are sorted by position from top to bottom. Thus, a potential result from the ×-combination of the expansion is {$Khoo_{pos1}$, $Guan_{pos2}$, $Soon_{pos3}$}, which is demonstrated by the traversal numbered 1 and is the 1st ranked result in Fig. 4. On the other hand, {$Khoo_{pos1}$, Soo_{pos2}, $Guan_{pos3}$} is the 4th ranked result in Fig. 4. As examples demonstrated in Section 4.2, many other forms, such as {Kho_{pos1}, Soo_{pos1}, Gun_{pos1}}, are illegal as no two tokens can occupy the same position, despite the fact that each token frequently occupies position 1 as they are popular Chinese first names. Similarly, expansion along X is legal, while along Y is illegal as two tokens occupy the same position, that is, position 2. The illegal paths are pruned away, and only the legal paths are carried over to the next step of the process.

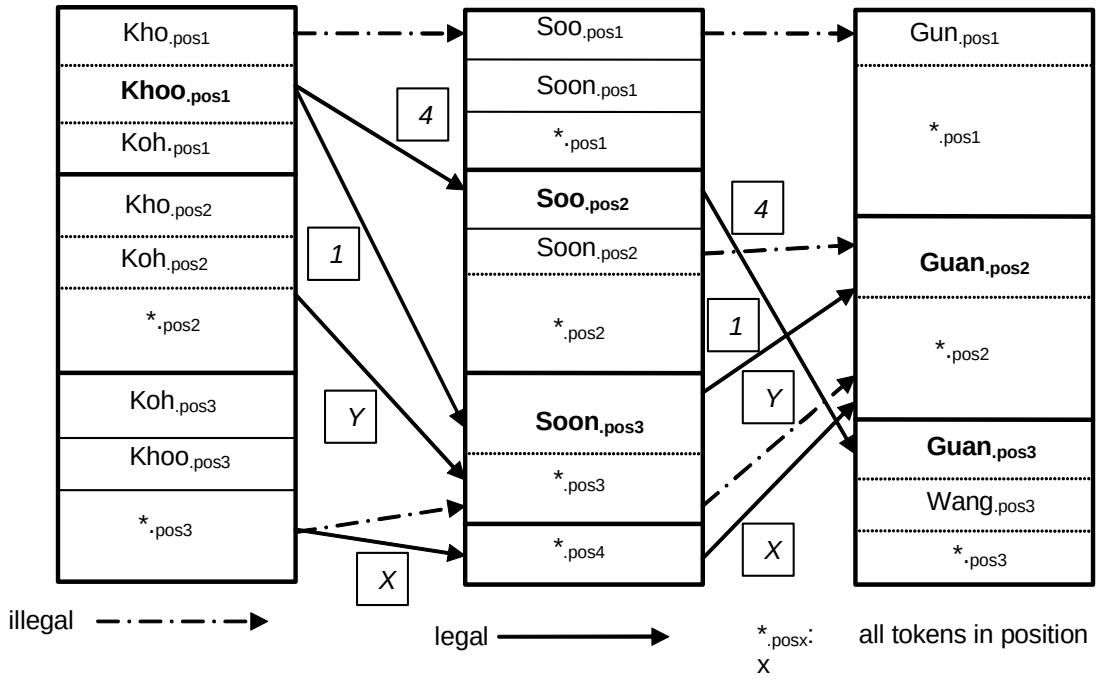

Fig 5. Illustration of position-as-attribute and position-transition-likelihood principles applied in the fuzzy matching process when "Kho Soo Gun" is issued as the search query

For each legal result $\{X_i, i=1 \text{ to } m\}$, where X_i is the name token at position i, the score is calculated as follows:

$$\prod_{i=1}^{i=m-1}\left[\frac{freq(pos_i(X_i))}{freq(X_i)} \times \frac{freq(pos_i(X_i), pos_{i+1}(X_{i+1}))}{(freq(pos_i(X_i)) + freq(pos_{i+1}(X_{i+1})))}\right] \frac{freq(pos(X_m))}{freq(X_m)} \quad (Formula\ 1)$$

where $freq(X_i)$ is frequency of token X_i, regardless of its position, and $freq(pos_i(X_i))$ is the frequency where token X_i occupies position i of a record; similarly to the joint frequency $freq(pos_i(X_i), pos_{i+1}(X_{i+1}))$. The rationale is that the more likely a position is associated with a token and a position transition is associated with a token pair, the more likely is a result containing the tokens and token pair is to be relevant, and the similarity is proportional to the normalized frequencies of the occurrences of such tokens and pairs in the name. The actual accumulation of postings still needs to be executed to enumerate the matching records.

Computationally, a further advantage is noted from applying the principles. The process illustrated in Fig. 3 actually prunes away those illegal results whose postings need not be accumulated, saving index access time. The reduction gained is generally in the order of the difference between $\frac{N!}{(N-M)!M!}$ and $\prod_{i=1}^{i=M} |A_i|$, where N is the potential positions (usually less than 10), M is the token length of the query name, and $|A_i|$ the sizes of each of the expanded token set.

5.1 Benchmark and Experimentation on FNS

FNS was developed into an industrial system soon after the initial invention and patenting process. A full scale evaluation is currently under way as part of the continuous R&D work on FNS. The results of this full scale experiment will be reported in a separate article. As with typical industrial deployment in the end-user environment, FNS, nonetheless, was subject to stringent benchmarking exercise before it was accepted, namely the User Acceptance Test (UAT). In the following, we shall review aspects of the UAT that provides some indication to FNS' quality of search. More industrial aspects of FNS deployment can be found on the website http://www.mustardtech.com

Normally, UAT's concern two distinct aspects of the quality of search: Performance and Functionality. In terms of search performance, it typically requires an average response time of 1-2 seconds given the total size of the database, which mostly ranges from a few hundred thousand records to a few million. With scalable computer architecture, the performance issue of FNS has become a scalability issue of its software system. As demonstrated in many of its deployments, FNS has been shown to be able to scale up its performance on Symmetric Multi-Processor (SMP) machines with response time inversely linearly proportional to the numbers of CPUs and RAMs. The 1-2 seconds response time is largely determined by the normal wait time before end-users become doubtful about obtaining the intended results of the search request. The 1-2 second limit may be adjusted according to the norms of expectation of different user groups.

The major focus of the UAT lies in the functionality of FNS; it contains test cases consisting of a list of name variants that may be used to query a target name. A system is considered satisfactory, in a similar sense to recall ratio, if more variants can be used to retrieve the target name. Typically, all the name variants listed in the test cases need to be able to retrieve the target name. The precision and ranking of the retrieved names are less of an issue as long as the target is included among the top N hits, where N can range from 7 to 20 depending on whether they can fit on a screen. As these test cases are not available for publication, in the following, a simulation of test cases was used. The sample space is derived from the names of the Faulty and Staff of School of Communication and Information (SCI), Nanyang Technological University (URL: http://www.ntu.edu.sg/sci/about/directory.html). This sample space has around 100 names and it reflects a typical distribution of name types in Singapore. From this sample space, four representative names were chosen as they

represent the four most common name types: Western, Chinese, Malay and Sanskrit/Indian name types. The final target names chosen are as follows:

- Name # 1: Abdus Sattar Chaudhry (Sanskrit/Indian)
- Name #2: Benjamin H. Detenber (Western)
- Name #3: Khoo Soo Guan, Christopher (Chinese)
- Name #4: Shahiraa Sahul Hameed (Malay)

Table 8 demonstrates the results of these test cases on FNS as compared to those on Google.

Table 8. Test Cases of FNS and Comparison with Google[5]

Name Variants Comparison	Listed Name	Initial/ Drop of Initial	Typographical Errors	Sound-alike	Combination of the above
Name # 1	*Abdus Sattar Chaudhry*	*Abdus S Chaudhry*	*Abdsu Sattar Chaudhry*	*Abdus Sattar Chawdhry*	*S Abdsu Chawdhry*
FNS	*Yes √*	*Yes √*	*Yes √*	*Yes √*	*Yes √*
Google	*Yes √*	*No X*	*No X*	*No X*	*No X*
Name # 2	*Benjamin H. Detenber*	*Benjamin Detenber*	*Benjmin H Detenber*	*Benjamin H Dedenber*	*Benjmin H Dedenber*
FNS	*Yes √*	*Yes √*	*Yes √*	*Yes √*	*Yes √*
Google	*Yes √*	*Yes √*	*No X*	*No X*	*No X*
Name # 3	*Khoo Soo Guan, Christopher*	*Khoo S G, Christopher*	*Khoo Soo Guan, Christoper*	*Khoo So Gun, Christopher*	*Christoper Khoo So Gun*
FNS	*Yes √*	*Yes √*	*Yes √*	*Yes √*	*Yes √*
Google	*Yes √*	*No X*	*No X*	*No X*	*No X*
Name # 4	*Shahiraa Sahul Hameed*	*Shahiraa S Hameed*	*Shahiraa Sahlu Hameed*	*Shahilaa Sahul Hamed*	*Sahlul Hamed Shahiraa*
FNS	*Yes √*	*Yes √*	*Yes √*	*Yes √*	*Yes √*
Google	*Yes √*	*No X*	*No X*	*No X*	*No X*

5 Two points to note: (1) All searches on Yellowpages return a unique record targeted by the name search, (2) Some of the initial search in FNS requires the specification of wild card operators, such as "Abdus S* Chaudhry".

All the names in the test cases are simultaneously listed on the SCI website and the Singapore Yellowpages, which runs on FNS. For comparison, Google's site specific search is included with the site parameter specified as "site:www.ntu.edu.sg/sci". As shown in Table 8, out of the 16 name variants tested, FNS retrieves all 16 of them. When the same search request is applied to Google, it retrieves 1 case. These cases are samples of four name variant types: Initial/Drop of Initial, Typographical errors, Sound-alike and Combination of the above. Thus it can be seen as indicative of main types of name variants that should be handled by a name search system. On the other hand, the cases are certainly not exhaustive and do not cover all the variants, to be investigated in a full scale study. As such, the extent that FNS satisfies overall information needs remain to be assessed quantitatively. Short of this full scale study, the comparative benchmark with the most widely used search engine, Google, is instructive. Google delivers a certain degree of fuzzy search by suggesting similar but different search strings that seem to reflect the frequency of co-occurrence. For instance, with a search on "Kandnsky Vassily" (a typographical error variant) , Google suggests it may be spelled as "Kandinsky Vassily," which is one of the more popular name transliteration of the renowned Russian/French artist – around 9,700 hits as of January 2006. On the other hand, given a deviant spelling "Kandinsky Wasily," it will suggest "Kandinsky Wassily," which is the most popular transliteration of the name – 539,000 hits. However, the two name variants: "Kandinsky Vassily" and "Kandinsky Wassily," are not suggested to be related in Google. This is in contrast to FNS' algorithm where the two variants are determined to be related to each other, and in our opinion, a more accurate suggestion. The comparison shown in Table 8 suggests that FNS is more effective than Google in name searching.

6. Conclusion

Technologists building theory-driven name search systems are confronted with two seemingly different alternatives: the Name-as-Language and the Name-as-Text approaches. The Name-As-Language approach treats names as word sequences generated by rule-governed grammar. As an alternative to the Name-As-Language view, the Name-As-Text approach assumes names are just records consisting of features derivable from name tokens. Based on Name-As-Language, a more data-driven method, called *position-as-attribute* principle, is proposed. The *position-as-attribute* principle regards name positions as attributes in name structures. The *position-transition-likelihood* principle, which is motivated by Name-As-Text, together with the *position-as-attribute* principle, is introduced to prune and verify the query expansion process. Thus, a theory-driven name search system, called Friendly Name Search (FNS), is built by combining the complementary advantages of both the Name-As-Language and Name-As-Text approaches to achieve effectiveness both in system development and quality of search. FNS has been applied to real world application in Singapore Yellowpages and many organizations in the public, banking and telecommunications sectors. Beyond an initial comparison on its

functionality with Google's, a full scale study on its search quality is being conducted currently and it will be reported in a separate article in the future.

7. References

[1] A. Spink, B. Jansen, and J. Pedersen, Searching for People on Web Search Engine. *Journal of Documentation* 60(3) (2004) 266-278.

[2] J.C. Hermansen, Automatic Name Searching in Large Data Base of International Names. Ph.D. Thesis, Georgetown University. (1985).

[3] S. L. Siegfried and J. Bernstein, Synoname: The Getty's New Approach to Pattern Matching for Person Names. *Computers and the Humanities* 25 (1991) 211-226.

[4] A. Golding, Pronouncing Names by a Combination of Rule-Based and Case-Based Reasoning, Ph.D. Thesis, Stanford University. (1991).

[5] E.M. Keen, Term Position Ranking: Some New Test Results, In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval.* Denmark. (1992)

[6] C. L. Borgman and S.L. Siegfried, Getty's Synoname and Its Cousins: A Survey of Applications of Personal Name-Matching Algorithms, *Journal of the American Society for Information Science* 43(7) (1992) 459-467.

[7] P.A. Hall and G. R. Dowling, Approximate String Matching, *Computing Surveys* 12(4) (1980) 381-402.

[8] G. J. Moore, Mechanizing a Large Register of First Order Patient Data, *Methods of Information in Medicine* 4(1) (1965) 1-19.

[9] K.G. Roughton and D.A. Tyckoson, Browsing with Sound: Sound-Based Codes and Automated Authority Control, *Information Technology and Library* 4 (1985) 130-136.

[10] D.W. Fokker and M.F. Lynch, Application of the Variety-Generator Approach to Searches of Personal Names in Bibliographic Database-Part 1. Microstructure of Personal Authors' Names, *Journal of Library Automation* 7(2) (1974) 105-118.

[11] R.L. Taft, Name Search Techniques, *New York State Identification and Intelligent System*, Albany (1970).

[12] I. Fellegi and A. Sunter, A Theory for Record Linkage, *American Statistical Association Journal* December 64 (1969) 1183-1210.

[13] U. Pfeifer, T. Poersch, and N. Fuhr, Retrieval Effectiveness of Proper Name Search Methods. *Information Processing & Management* 32(6) (1996) 667-679.

[14] P. Thompson and C. Dozier, Name Searching and Information Retrieval. In *Proceedings of Second Conference on Empirical Methods in Natural Language Processing*, Providence, Rhode Island USA (1997) 134-140.

[15] G. Navarro, R. Baeza-Yates, J.M Azevedo-Arcoverde, Matchsimile: A Flexible Approximate Matching Tool for Searching Proper Names, *Journal of the American Society for Information Science and Technology* 54 (1) (2003) 3-15.

[16] G. B. Beli and A. Sethi, Matching Records in a National Medical Patient Index, *Communication of the ACM* 44(9) (2001) 83-88.

[17] B. Stalls, K. Knight, Translating Names and Technical Terms in Arabic Text, In *Proceedings of the COLING/ACL Workshop on Computational Approaches to Semitic Languages* (1998) 34-41.

[18] P. Virga and S. Khudanpur, Transliteration of Proper Names in Cross-Language Applications, In *Proceedings of SIGIR 2003*, Toronto, Canada, (2003) 365-366.

[19] A. Pirkola, J. Toivonen, H. Keskustalo, K. Visala, and K. Järvelin, Fuzzy Translation of Cross-Lingual Spelling Variants. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval. Toronto, Canada* ((2003) 345-352.

[20] E. Brill, G. Kacmarcik and C. Brockett, Automatically Harvesting Katakana-English Term Pairs from Search Engine Query Logs, In *Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium*, (2001) 393–399

[21] L. S. Larkey, N. Abdul Jaleel and M. E. Connell, What's in a Name? Proper Names in Arabic Cross-Language Information Retrieval. CIIR Technical Report, IR-278. (2003)

[22] F. Patman, and P. Thompson, Names: A New Frontier in Text Mining. In Hsinchun Chen, Richard Miranda, Daniel D. Zeng, Chris Demchak, Jenny Schroeder, Therani Madhusudan, Intelligence and Security Informatics. First NSF/NIJ Symposium, ISI 2003, Tucson, AZ, USA, Proceedings, Springer Verlag Heidelberg. (2003) 27–38.

[23] R. Oshika, F. Machi, B. Evans and J. Tom, Computational Techniques for Improved Name Search. *Proceedings of the second conference on Applied natural language processing*. Austin, Texas. (1998) 203-210

[24] A. Bosch and W. Daelemans, Data-Oriented Methods for Grapheme-to-Phoneme Conversion, In *Proceedings of the Sixth Conference of the European Chapter of the ACL, Utrecht*, (April, 1993) 45-53.

[25] K. Church, A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text, In *The Second Conference on Applied Natural Language Processing, ACL, Austin, Texas.* (1988) 136-143.

[26] S. DeRose, Grammatical Category Disambiguation by Statistical Optimization, *Computational Linguistic 14*(1) (1988) 31-39.

[27] W. S. Wong and M. C. Chuah, A Hybrid Approach to Address Normalization. *IEEE Expert* 9(12) (1994) 38-45.

[28] Y. Qiu and H. P. Frei, Concept Based Query Expansion, In *Proceedings of the Sixteenth Annual International ACM SIGIR Conference*, Pittsburgh, PA, USA, June-July, (1993) 160-169.

[29] Y-C. Wang, J. Vandendorpe, and M. Evens, Relational Thesauri in Information Retrieval, *Journal of the American Society for Information Science* 36(1) (1985) 15-27.

[30] K.W. Church and P. Hanks, Word Association Norms, Mutual Information and Lexicography, In *Proceedings of the 27th Annual Conference of the Association of Computational Linguistics*, Association for Computational Linguistics, New Brunswick, NJ, (1989) 76-83.

[31] P. Anick and S. Vaithyanathan, Exploiting Clustering and Phrases for Context-Based Information Retrieval. In *Proceedings of the 20th Annual International ACM SIGIR Conference,Philadelphia* (1997) *USA* 314-323

[32] H. Chen and K. J. Lynch, Automatic Construction of Networks of Concepts Characterizing Document Databases, *IEEE Transactions on Systems, Man, and Cybernetics* 22(5) (1992) 885-902.

[33] M. Buckland and F. Gey, The Relationship between Recall and Precision, *Journal of the American Society for Information Science* 45(1) (1994) 12-19

[34] G. Salton, Developments in Automatic Text Retrieval. *Science* 253 (1991) 974-979.

[35] G. Salton and C. Buckley, Term-Weighting Approaches in Automatic Text Retrieval, *Information Processing and Management* 25(5) (1988) 513-523.

[36] E.M. Keen, The Use of Term Position Devices in Ranked Output Experiments, *The Journal of Documentation* 47(1) (1991) 1-22.

[37] E.M. Keen, Term Position Ranking: Some New Test Results, In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval.* Denmark. (1992) 66-79.

[38] W. B. Croft, H. Turtle, and D. Lewis, The Use of Phrases and Structured Queries in Information Retrieval, In *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval.* Chicago, USA. (1991) 32-45.

[39] J. Fagan, The Effectiveness of a Nonsyntactic Approach to Automatic Phrase Indexing for Document Retrieval, *Journal of the American Society for Information Science* 40(2) (1989) 115-132.

[40] M. Mitra, A. Singhal, and C. Buckley, Improving Automatic Query Expansion. In.*Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval.* Melbourne, Australia (1998) 206-214.

[41] M. Hearst, Improving Full-Text Precision on Short Queries using Simple Constraints. In *Proceedings of SDAIR.* Las Vegas, USA. (1996) 217-228.

[42] P.H.J., Wu, Z. Q. Shen, S. Guo, P. S. Lim, T. J. Chng, C. J. Chong and H. B. Low, Technologies in Meta-Information Management and Service. In *Proceedings of the joint Pacific Asian Conference on Expert Systems and Singapore International Conference on Intelligent Systems, Singapore* (1997) 711-720.