Matching Word-Order Variations and Sorting Results for the iEPG Data Search

Denis Kiselev, Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan

Rafal Rzepka, Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan

Kenji Araki, Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan

ABSTRACT

This paper describes using a finite-state automaton (FSA) to retrieve Japanese TV guide text. The proposed FSA application can be considered novel due to lack of research on the subject. The automaton has been implemented for matching and extracting all possible combinations of search query words in all possible word orders that may be present in the TV guide text. This implementation also sorts the extraction results by analyzing word semantic features (such as "being an object" or "being a property of an object"). The present paper also proposes a search system using the above implementation and compares it with a baseline system that matches query words (of multi-word queries) in exactly the same and exactly the opposite word orders only. Both systems use morphological parsing and apply a stop list to the query. A multi-parameter evaluation has shown advantages of the proposed system over the baseline one.

Keywords: *Electronic Program Guide (EPG), Finite-State Automaton (FSA), Information Retrieval, Lexical Semantics, Morphological Parsing, Natural Language Processing (NLP)*

MOTIVATION FOR THIS RESEARCH

Japanese is written without spaces between words. That means a search system processing this language needs to "know" what character strings are words, or at least where character strings that could be words start and end. It is even better if a system attempts to find out what those words, or groups of characters, may mean. The same is true for searching the Japanese language iEPG (Internet Electronic Program Guide or, simply, Web pages saying when, what programs are shown on TV).

It can be concluded from the output of search systems available on major Japanese iEPG websites[1] that those systems most likely apply the direct matching technique to the query,

DOI: 10.4018/ijmdem.2014010104

treated by them as a character string. In other words, they most likely match the search phrase without segmenting it into words (i.e. without morphological parsing and inserting spaces at word boundaries).

Kiselev et al. (2013) suggested improvements to that technique and proposed an iEPG search system utilizing morphological parsing and the core meaning analysis for matching the search query with the TV guide text.

The above authors also demonstrated how using that system could improve search results, however matching query words in all possible orders was left for future work.

The system proposed by the above authors can match query words (provided the query has two or more of them) in exactly the same or exactly the opposite orders only (ibid.). For two-word queries "exactly the same" and "exactly the opposite" are all the possible word order options, however there are more options for longer queries. Thus, the system will successfully match text with "観光地は人気で綺麗 ([*kankouchi wa ninki de kirei*] the sightseeing spot is popular and beautiful)"[2] in response to the query "綺麗で人気な観光地 ([*kirei de ninki na kankouchi*] a beautiful and popular sightseeing spot)", but will not match the same text in response to "人気で綺麗な観光地 ([*ninki de kirei na kankouchi*] a popular and beautiful sightseeing spot)".

This ability to express (practically) the same meaning using the same words in various orders is described as a characteristic feature of "context-free languages", i.e. ones allowing more flexible word combinability, by Maruoka (2011). The order flexibility in Japanese word combinations is illustrated in terms of the "context-free grammar" and NLP (Natural Language Processing) by Tanabe, Tomiura and Hitaka (2000).

Implementing a system capable of matching query words in all possible orders characteristic of the Japanese language, has been the primary motivation for the research described in this paper. The system proposed by Kiselev et al (2013) (mentioned earlier in this section) has been used as a baseline.

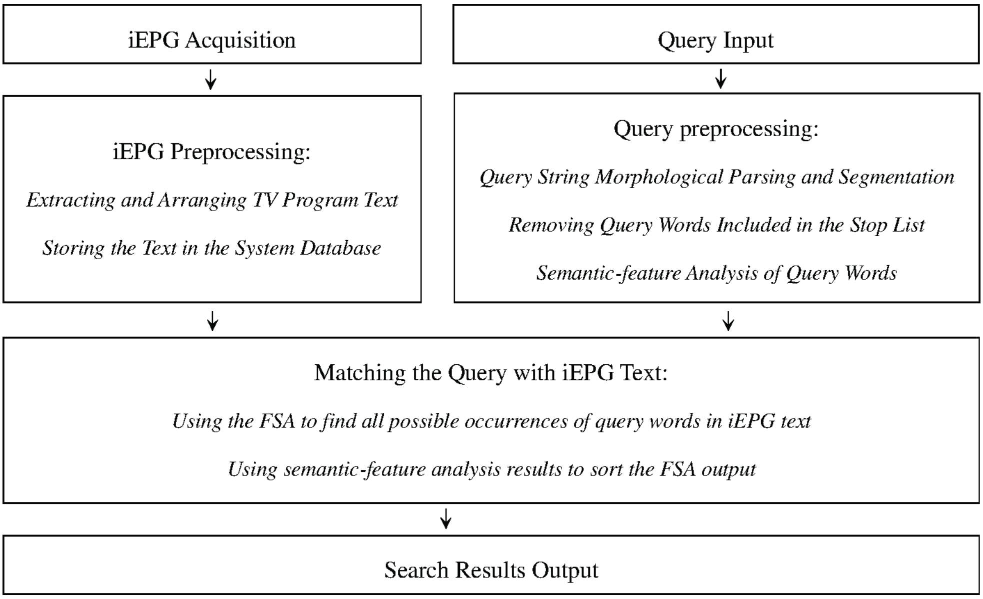
It should be noted that both the baseline and proposed system implementations are essentially different form large search engines, such as Google. First, large search engines retrieve web documents, such as websites and parts of them, whereas the proposed and the baseline implementations retrieve pieces of text that describe TV programs. To do so, the implementations do not require indexing millions of web documents (the way a Google webpage[3] says it does) and do not need any corpora, such as the approximately 24-GB large Google N-gram Corpus described by Lin et al (2010). Because of their size the implementations could be used locally as, say, internal search systems for TV sets. It seems unlikely that, for instance, the Google search engine can be used in the same way. It has been our purpose to develop the search system for the TV program guide by taking into account the above peculiarities of this task.

INPUT-OUTPUT FLOW OF THE PROPOSED SYSTEM

This section contains a concise flow description. Sections that follow explain flow stages in more detail.

Figure 1 shows the major flow stages. The initial stage is TV guide data acquisition. Once the data are offloaded from a Web source by the system and a query is input, the guide and the query are preprocessed, i.e. changed into the format suitable for matching. At the preprocessing stage, the query string first undergoes morphological parsing, stop-listing and semantic-feature analysis (the analysis is described in C. Semantic-feature Analysis of Query Words). The query is then matched with the TV guide text by means of a finite-state automaton (FSA). The semantic-feature analysis results are used to sort (according to possible relevance to the query) TV guide data

Figure 1. Flow outline


iEPG Acquisition
Query Input
iEPG Preprocessing:
Extracting and Arranging TV Program Text
Storing the Text in the System Database
Query preprocessing:
Query String Morphological Parsing and Segmentation
Removing Query Words Included in the Stop List
Semantic-feature Analysis of Query Words
Matching the Query with iEPG Text:
Using the FSA to find all possible occurrences of query words in iEPG text
Using semantic-feature analysis results to sort the FSA output
Search Results Output

(that match the query). Finally, the results of this procedure are output.

iEPG ACQUISITION

In order to explain how the proposed search system acquires and further processes the TV guide there is a need to explain what exactly iEPG is and how it is used.

Nowadays for the convenience of the spectator, TV program guide text comes in the searchable electronic format. Such data are referred to as EPG (Electronic Program Guide). In Japan the guide can be browsed through built-in features of most television sets, as well as by using a PC interface to access the data in the WWW. For this Internet-based TV guide version, a term "iEPG" (meaning, as mentioned in the initial section of this paper, "Internet Electronic Program Guide") has been coined. Examples of Japanese iEPG sites can be found at URLs given in footnote 1.

Data in the online TV guide are grouped, each group describing a single program. The description natural language text includes such information as the name for the TV channel broadcasting the program, the broadcasting date and time, the program genre, cast and contents from a word or two to about a paragraph in length. More details on the Japanese iEPG format, including examples, are given by Yamasaki, Manabe & Kawamura (2008). The iEPG data format used by the system this paper proposes is described in the section iEPG PREPROCESSING.

Procedures implemented at the acquisition stage are listed below:

1. Locating (Java) scripts for loading detailed TV program data on the iEPG Web page;
2. Using the located scripts to generate Ajax requests for the detailed TV guide data;
3. Using the requests to offload the data in JSON format.

Japanese iEPG websites normally display two types of the TV Guide. The first type is concise, the second is detailed. Concise TV program descriptions have clickable buttons available to the human user (and programming scripts to be used by the machine) for loading detailed data, i.e. displaying these data to the user. The proposed system uses the scripts to offload the detailed data.

After accessing the iEPG Web page and finding scripts, the system makes server requests for the data using the Ajax technology. Eichorn (2006) explains in detail how that technology can be used for a variety of purposes.

The requested data is offloaded for each TV program in the JSON format, i.e. as natural language text with various metadata such as tags indicating the content type for different parts of that text. For example, program title text with a tag indicating that this text is a program title is offloaded for each TV program. This format is useful for data structuring and extraction as it explicitly indicates where, what kind of text can be found. A way the JSON technology can be used for retrieving data as “key-value pairs” (similar to the pair including the program title and the tag in the example above) is described by Tummarello et al (2010).

At this time the proposed system downloads iEPG data for the eight TV channels that are broadcast terrestrially in Sapporo, Japan. The data is downloaded for the current date and seven days ahead and includes descriptions for about two thousand TV programs.

iEPG PREPROCESSING

Below is a list of procedures implemented at the iEPG preprocessing stage:

1. Extracting natural language text for TV programs from the JSON data structure;
2. Arranging the text;
3. Storing each TV program text as a separate item in the system database.

Natural language text is extracted from the tag-text pairs described above and tags (such as “program title” of “program genre”) indicating contents of text chunks are used to arrange them. The chunks are arranged in the order shown below. Each item of the following list is numbered according to the order and gives the content type of each text chunk:

1. Name of the TV channel broadcasting the program;
2. Broadcasting type (e.g., terrestrial), channel number;
3. Starting date and time, ending date and time for the program;
4. Program title;
5. Program subtitle;
6. Program content summary;
7. Detailed program content explanation;
8. Program genre;
9. Program cast or list of participants.

The above arrangement follows the style iEPG is listed on Japanese websites. If some of the above items (such as “program subtitle” and “program content summary”) are absent from the actual TV guide, the proposed system arranges the available items in the above order without leaving any items blank. Arranged in this way, the text for each TV program is stored as a separate item in the system database. The database items are matched one by one with query words at the matching stage.

As mentioned above, sometimes the actual TV guide does not have any text for some of the above list items (because the TV guide is arranged in this way by the broadcaster, the Web administrator or both.) If an item is missing, i.e. there is blank space instead of it, the system ignores the blank space and goes on to the next item for which text is available. For instance, if text for item ⑤ (“program subtitle” of the above list) is missing from the TV guide, the system stores the program text without the missing subtitle in the database and does not leave any blank space in place of the subtitle. As the system does not ignore any TV program text present in the TV guide and does not process what is absent from the guide, the search accuracy is unaffected.

QUERY PREPROCESSING

The following procedures are implemented to preprocess the search query:

1. Query string morphological parsing and segmentation;
2. Removing query words included in the stop list;
3. Semantic-feature analysis of query words;
4. Storing the analysis results to be used at the matching stage.

Query String Morphological Parsing and Segmentation

Differently form n-gram-based statistical approaches to information retrieval, e.g. one proposed by Miller et al (2006), we emphasize taking into account such aspects as the morphological structure (of the Japanese language).

The proposed system does not use the n-gram model for segmenting the query, i.e. splitting the query string into separate words. The statistical approach to query segmentation, such as one described by Wang et al (2010), involves matching a character string with strings repeatedly found in a corpus (e.g., one containing a large number of n-grams). Shorter strings that often match are considered to be separate words and are singled out. Wang et al (2010) describe using Microsoft Web N-gram Corpus for segmenting the query in this way.

We have preferred morphological parsing to purely statistical segmentation because n-gramming a character string does not take into account the morphology, i.e. rules governing the way words are constructed from morphemes, such as word stems, endings etc.

The proposed system segments the query by means of a morphological parser JUMAN[4]. The parser is for processing the Japanese language. It is described by Kawahara and Kurohashi (2013). The reasons we chose the parser are that it is one of the fastest according to Ptaszynski et al (2012), and that it takes into account not only statistics but also Japanese morphology rules.

Before feeding the query to the parser, the query character string is checked to ensure it is encoded in uft-8 and all half-width characters are substituted with full-width ones (which is required for using the parser). To divide the string into segments, JUMAN analyzes such Japanese language features as parts of speech combinability and inflections. Along with the segmented string, JUMAN output has other information such as tags telling what part of speech each segment (i.e. a character string JUMAN considers a word) is. Words and their tags are taken out of the output and used for the semantic-feature analysis described in one of the following subsections.

Removing Query Words Included in the Stop List

The proposed system checks each word of the query against a stop list of words with little conceptual meaning and of little value from the information extraction viewpoint. Such words are removed from the query.

In existing research various kinds of stop lists and their applications have been considered. For English, Hiemstra and de Jong (2001) suggest removing words with little conceptual meaning (such as “a”, “the” and “it”) from the query as well as from the indexed text that is searched. Fukuta et al (2002) describe a system (for processing the Japanese language) that lists words of no potential interest to the user as stop list items.

We suggest using a stop list for multiple reasons. That is, some parts of a word (and sometimes the whole word) for structural, semantic and pragmatic reasons can be omitted or substituted with others with no change to the meaning. The system we propose, as mentioned above, detects and discards such words and morphemes. Table 1 lists them and gives examples of the way they may be used in a search query. In Table 1 query examples,

Table 1. The stop list

No.	Stop List Entry	Entry Classification	Entry Use in a Query
①	は [*wa*]	a particle	料理は美味しい [*ryouri wa oishii*] the food is tasty
②	が [*ga*]	a particle	温泉がある地域 [*onsen ga aru chiiki*] area with hot spring (spa)
③	の [*no*]	a particle	札幌の天気 [*sapporo no tenki*] Sapporo weather
④	な [*na*]	a pre-noun adjectival ending that can be substituted with "い[*i*]" with no change to the word meaning	小さな旅 [*chiisana tabi*] little trip
⑤	い[*i*]	an adjective ending that can be substituted with "な[*na*]" with no change to the word meaning	小さい町 [*chiisai machi*] small town
⑥	ある [*aru*]	a verb	温泉がある地域 [*onsen ga aru chiiki*] area with hot spring (spa)
⑦	いる[*iru*]	a verb	セレブがいる風景 [*serebu ga iru fuukei*] scene with celebrity

English articles are sometimes omitted to save the space and preserve the query style. In the "Entry Use in a Query" column and other parts of this paper that follow the table, stop list items written in Japanese and their transliterations are underlined for clarity. As mentioned earlier, all transliterations are italicized.

The stop list currently has entries of seven types. The reasons they have been included in the list are explained below. The inclusion decision is based on the human analysis of the search results for multiple queries with the stop list items as parts of them.

It can be said that the particles "は[*wa*]" and "が[*ga*]" are interchangeable with no dramatic change to the meaning. In other words, the particles could be roughly compared to the English definite and indefinite articles that convey definiteness nuances without changing the lexical meaning of what they modify. Including "は[*wa*]" or "が[*ga*]" in the query (like in the example for item ① in Table 1) as a mandatory match, would mean making the search system look for something not really needed for retrieving the meaning searched for. Moreover, if the system uses direct matching techniques, as the ones for the iEPG site examples mentioned in footnote 1 most likely do, for instance, "料理が美味しい ([*ryouri ga oishii*] food is tasty)" will not match "料理は美味しい ([*ryouri wa oishii*] the food is tasty)" although the two phrases mean practically the same.

The stop list item "の [*no*]" is often used as a possessive particle. According to a Japanese to English dictionary[5], it also can express the idea that "something is a location for something else" or "that something is the site of a certain action". A different Japanese to English dictionary[6] suggests that phrases in which "の [*no*]" is used in a non-possessive meaning be reworded to avoid using it. In many Japanese texts, typically technical, the particle is simply omitted. In fact, in example ③ above, "の [*no*]" (used in a non-possessive meaning) can also be omitted. Thus searching for it is unnecessary.

The items "な[*na*]" and "い[*i*]" can be considered variant endings. The same stem can have either of them with no practical change to the meaning. It is common knowledge that, for instance, the prenominal adjectival "小さな [*chiisana*]" can become "小さい [*chiisai*]" and the meaning of both is practically the same, "small".

If direct matching is used, a query with the former will not match the text with the latter and vice versa. For search precision reasons the system filter for "い[*i*]" endings is limited to those adjectives that have "な[*na*]" pre-noun adjectival counterparts. Counterparts from Sanseido Web Dictionary (indicated in footnote 5) are used for the filter.

The items "ある [*aru*]" and "いる [*iru*]" are verbs denoting the presence of an inanimate or animate object respectively. As other verbs referring to a certain object normally presuppose the presence of that object, removing "ある [*aru*]" and "いる [*iru*]" from the query can broaden the search scope. Thus if "ある [*aru*]" and "いる [*iru*]" are removed from a query that includes these verbs with their subjects, the query will match a text having the same subjects and other verbs, including those presupposing "ある [*aru*]" or "いる [*iru*]" meanings. Such broadening of the scope, however, also can result in retrieving verbs with the opposite meaning, "the absence". As it is a common sense matter that a user looking for something or somebody present somewhere also might be interested in the text about the same entity absent from some place, "ある [*aru*]" and "いる [*iru*]" are included in the stop list.

Semantic-Feature Analysis of Query Words

This subsection gives the essence and theoretical background of what we refer to as "semantic-feature analysis". It also explains how and why the analysis has been implemented.

Existing research demonstrates that morphological features of a word to some extent determine its semantic features. That is, if a word has certain morphemes, it belongs to a certain part-of-speech and basic meaning category. For instance, the suffix *–ist* of the word *guitarist* makes it a "denominal person noun" (Lieber, 2004). Another research states that a word of a language has the "semantic core" also referred to as the "semantic prime" (Goddard, 2002). For example, semantic primes for nouns are classified as "substantives" and those for adjectives as "determiners" (ibid.). Moreover, according to Wierzbicka (1996) semantic primes are "universal", i.e. present in multiple languages. The method we propose focuses on two types of semantic primes, i.e. *the object* and *the property-of-an-object* meaning features. The former is characteristic of nouns, the latter of adjectives.

In other words, by the semantic features the proposed system analyzes we mean the semantic primes discussed above. By the semantic prime for the noun we mean the fact that nouns signify objects, and by the semantic prime for the adjective that adjectives signify properties of objects.

The proposed system bases its analysis of the sematic features of search phrase words on the morphological analysis and part-of-speech tagging. Using the part-of-speech tags (described in A. Query String Morphological Parsing and Segmentation) the system attempts to make a judgment on the following three aspects:

1. Whether the user is searching for nouns, i.e. words meaning objects;
2. Whether the user is searching for adjectives, i.e. properties of objects;
3. Whether the user is searching for words different from the above.

When searching TV program data the system performs the semantic-feature analysis for the following reasons. An existing research demonstrates that nouns "constitute over 70% of query terms"[7] (Barr et al., 2008).

Moreover, nouns used together with adjectives are "common need information clusters" in English queries for multiple search engines

(Baeza-Yates et al., 2005). Another research demonstrates that nouns, such as proper nouns, are numerous in Japanese search queries (Arita et al., 2007).

To sum it up, the proposed system looks into the universally present core meaning of the query to find object and property-of-an-object features. If found, words with these meaning features become the focus of the analysis because nouns form the majority of query terms and noun-adjective clusters are common in queries for multiple search engines.

We believe that the proposed analysis technique could be used not only for Japanese but also for other languages. The fact that object and property semantic features are "universal", as stated above, justifies this belief.

MATCHING THE QUERY WITH iEPG TEXT

The system offloads and preprocesses iEPG data for the current date and seven days ahead. This eight-day data is as much as one can presently obtain from the major Japanese iEPG websites (mentioned in the section MOTIVATION FOR THIS RESEARCH). At the matching stage the query words are matched with text for one TV program at a time. That is, instead of matching the query with all the eight-day TV guide text at once, the system takes programs one by one to match the text for a single TV program with the query. D'hondt et al (2010) state that dividing long text into segments can facilitate automated parsing of the text. Moreover, a parser may fail if that is not done (ibid).

Reasons that dividing the TV guide text by program has been implemented for the proposed system, deal with the matching automaton load, the iEPG data peculiarity and the search precision.

Splitting the guide text by program definitely puts lighter loads on the automaton than making it match the entire eight-day guide at once.

The iEPG peculiarity consists in the fact that each program description in the guide can most-likely be considered semantically independent. In other words, information in one program description is most unlikely to refer to another program description.

This peculiarity can affect the search precision. For instance, retrieving two program descriptions one containing "Tokyo food", the other "Kyoto weather" in response to the query "Tokyo weather" is imprecise as the user is looking for "weather" (but not in Kyoto) and not for "food".

To illustrate this precision issue in more detail, let us consider two semantically independent TV program descriptions, one containing the phrase "Tokyo food" and the other the phrase "Kyoto weather". A query "Tokyo weather" matches the word "Tokyo" in the first program description and the word "weather" in the second one, so both descriptions could be retrieved as search results. However this is definitely imprecise because the two program descriptions are semantically independent and the user is looking for "weather" (but not the weather in Kyoto) and not for "food".

The following procedures are implemented at the matching stage:

1. Using the FSA to find all possible occurrences of query words in iEPG text;
2. Using semantic-feature analysis results to sort the FSA output.

USING THE FSA TO FIND ALL POSSIBLE OCCURRENCES OF QUERY WORDS IN iEPG TEXT

The FSA has been implemented for the purpose of solving the following problem. The problem consists in the need to extract TV guide text with all possible combinations of search query words in all possible word orders, with or without other words between the query words. Below is a mathematical model for the implemented FSA. Figure 2 demonstrates iteration cycles of the automaton. The model and the figure follow the conventional style of the automata theory literature (Aziz et al., 2004; Kumar, 2011).

The following equation shows the way the implemented FSA can be modelled mathematically:

$$M = (Q, I, \partial, W)$$

1. M represents the matching automaton implemented;
2. Q is the number of states, i.e. the number of times the automaton processes each word in the TV guide;
3. I is the set of all words in the TV guide text;
4. ∂ represents the transition function defined as $\partial(0, i_0) \rightarrow i_{n.}$ In $(0, i_0)$ 0 is the initial transition state at which the FSA attempts to match the very first word represented by $i_{0.}$ Each word the automaton attempts to match next is represented by i_n;
5. W is the set of accept states, i.e. the states at which query words match ones in the TV guide text. All the matches are sorted the way described in the following section.

As demonstrated in Figure 2, the automaton starts matching from the first TV program text ("p 1" in Figure 2). It takes the first query word ("w 1") and repeatedly matches it with the text in order to find all occurrences of the word.

The same is repeated for each consecutive word of the query ("w 1+n"). The automaton then goes on to each consecutive program text ("p 1+n") and repeats the same matching procedures.

Regular expressions and other constructs available in Perl (the programming language) are used to implement the automaton.

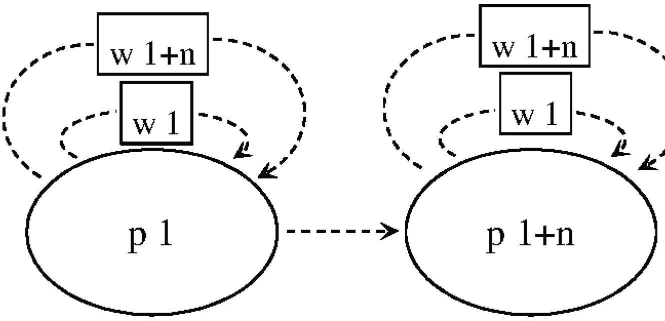
USING SEMANTIC-FEATURE ANALYSIS RESULTS TO SORT THE FSA OUTPUT

According to existing research, finite-state automata are implemented for such tasks as information extraction based on matching words (Smrz & Schmidt, 2009) or matching groups of words (Kwak et al., 2011). The FSA for the proposed system not only matches words but also groups the matching results according to semantic features. As explained earlier, the implemented automaton uses query words output resulting from the segmentation performed on the query string by the JUMAN parser.

McCandless and Hatcher (2010) state that search results can be grouped according to relevance by means of analyzing how well strings match, or by means of analyzing indexes attached to strings.

Our approach is more similar to the latter analysis, provided part-of-speech tags attached to words by JUMAN are looked upon as indexes. The automaton used in the proposed system "knows" the semantic-feature analysis

Figure 2. Iteration cycles of the matching automaton



results based on JUMAN tags. In other words it "knows" whether it is matching words meaning objects (e.g., "food"), words meaning properties of objects (e.g., "tasty"), or words without these semantic features.

At the end of each matching cycle (i.e. after the automaton is through with processing each TV program) the above "knowledge" is used to sort matching results.

The implemented filters put the results (chunks of program text, described in the section iEPG Processing) into the following six groups. The group contents (if any) are output for the user in the order shown below. The semantic features that words in each group must possess and other grouping criteria are as follows:

1. TV program text with one or more groups of words meaning objects and properties;
2. TV program text with two or more words that mean objects that are not the same; this text does not include words meaning properties;
3. TV program text with one or more words unknown to JUMAN but without words JUMAN has analyzed as objects or properties;
4. TV program text with one or more of the same object and no properties;
5. TV program text retrieved in response to one-word queries;
6. TV program text that does not meet the criteria in 1 through 5.

As explained above, the proposed system not only analyzes sematic features but also attempts to find out if words possessing them are the same.

In accordance with the above grouping criteria, the proposed system does not look upon the number of times the same word appears in the program text as the key factor in judging about the search result relevance. It rather gives priority to groups of words meaning objects and their properties, and groups of ones meaning two or more different objects, provided such words are present in the search query.

EVALUATION AND COMPARISON OF THE PROPOSED AND BASELINE SYSTEMS

This section first explains the major differences and similarities of the baseline and proposed systems, and describes the test data used for the evaluation. The section then provides recall, precision and other evaluation results for the two systems.

As stated in the section MOTIVATION FOR THIS RESEARCH, the system proposed by Kiselev et al. (2013) is used as a baseline. The major difference between the baseline and proposed systems is as follows. The baseline system matches query words of multi-word queries in exactly the same and exactly the opposite word orders only, whereas the proposed one matches query words in all possible combinations and orders.

The baseline system uses semantic-feature analysis to locate words with object and property-of-an-object features in the query. Such words are matched mandatorily while other words are matched optionally. The proposed system uses the semantic-feature analysis to sort the automaton output the way explained in the section USING SEMANTIC-FEATURE ANALYSIS RESULTS TO SORT THE FSA OUTPUT.

Both systems use morphological parsing and apply the same stop list to the query. For higher recall, the baseline system has been programmed to retrieve TV guide text in which zero or more other words may appear between the words used in the query.

TV guide text for eight days and seven channels broadcast terrestrially in Sapporo, Japan has been used as test data for evaluation. The data includes text for over 2,100 TV programs.

Search queries used for evaluation have been contributed by ten adults uninvolved in the present research. The individuals were asked to provide queries they would use for searching the TV program guide. They were also asked to use as many stop list items (listed in Table 1) as possible in the queries.

Thirty queries have been used for evaluation. Twenty-five of them include two or more words, five are single-word queries.

Table 2 compares evaluation results for the two systems. The formulas used to calculate recall and precision follow the traditional pattern described by Jizba (2007). For F-measure calculation we have used the formula given by Sasaki (2007).

The Relevant Results Retrieved column shows how many relevant program descriptions (i.e. chunks of program text, described in the section iEPG PREPROCESSING) were retrieved by each of the systems. Our judgment criteria regarding the relevance are explained below.

First, we have considered the following guidelines for relevance judgment found in an information retrieval source:

> *... it is generally easier for people to decide between at least three levels of relevance, which are definitely relevant, definitely not relevant, and possibly relevant. These can be converted into binary judgments by assigning the possibly relevant to either one of the other levels ...* (Croft et al., 2009, p. 8)

Speaking the language of this source, we have assigned "possibly relevant" to "relevant", however with a certain amount of caution. When examining search results we did not use the presence of query words in the search result text as the only criterion for relevance (or possible relevance). The context was considered along with the word presence. The following example (rendered into English) demonstrates the importance of the context. The program text "this place is not a resort" retrieved in response to the query "popular resort" falls, in our opinion, into the "irrelevant" category.

Generally speaking, from comparing the evaluation results (as per Table 2) it is clear that they are considerably better for the proposed system. Comparing the evaluation results in Table 2 (by subtracting respective values for the baseline system from those for the proposed one) demonstrates the following increase values for the proposed system:

1. Increase in the number of relevant results retrieved: 1260
2. Average recall increase: 0.61
3. Average precision increase: 0.33
4. Average F-measure increase: 0.46

Moreover, due to sorting search results by semantic features (explained in the section USING SEMANTIC-FEATURE ANALYSIS RESULTS TO SORT THE FSA OUTPUT), the proposed implementation has listed search results with higher semantic value closer to the top of the list. As the baseline implementation does not have the capability to do so, the result sorting is another advantage of the proposed implementation.

CONCLUSION AND FUTURE WORK

A system for searching the Japanese TV program guide has been implemented. The system uses

Table 2. System evaluation results

	Relevant Results Retrieved	Average Recall	Average Precision	Average F-measure
Proposed System	1483	0.91	0.72	0.76
Baseline System	223	0.30	0.39	0.29

the finite-state automaton in order to match query words in all possible combinations and orders, and sorts search results by their sematic features.

The proposed system has been evaluated by comparing its performance with that of a baseline system. The baseline system matches query words in exactly the same and opposite orders only, it is allowed that zero or more words between the query words appear in the iEPG text.

Both systems use stop-listing, morphological parsing and semantic-feature analysis to preprocess the search query.

From the multi-parameter evaluation, it can be concluded that the proposed system produces considerably better search results.

In the future we intend to incorporate syntactic parsing into the proposed system. The fact that the implemented automaton does not "see" syntactic relations among the words that it matches, can be considered a limitation of the current implementation. It is our intention to find out whether using syntactic parsing for the current system can further improve search results quality.

REFERENCES

Arita, I., Kikuchi, H., & Shirai, K. (2007). Word clustering using concurrent search queries, IEICE technical report, NLC. *Language Understanding and Models of Communication*, *107*(158), 115–120.

Aziz, A. D., Cackler, J., & Yung, R. (2004). *Automata theory*. Eric Roberts' Sophomore College, Stanford University. Retrieved November 9, 2013, from http://www-cs-faculty.stanford.edu/~eroberts/courses/soco/projects/2004-05/automata-theory/basics.html

Baeza-Yates, R., Hurtado, C., Mendoza, M., & Dupret, G. (2005). Modeling user search behavior. In *Proceedings of the Third Latin American Web Congress (LA-WEB '05)*.

Barr, C., Jones, R., & Regelson, M. (2008). The linguistic structure of English web-search queries. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics* (pp. 1021-1030).

Croft, B., Metzler, D., & Strohman, T. (2009). *Search engines: Information retrieval in practice*. Addison-Wesley.

D'hondt, E., Verberne, S., Oostdijk, N., & Boves, L. (2010). Re-ranking based on syntactic dependencies in prior-art retrieval. *Information Foraging Lab, 23*.

Eichorn, J. (2006). *Understanding AJAX: Using JavaScript to create rich internet applications*. Prentice Hall PTR.

Fukuta, H., Matsuo, Y., & Ishizuka, M. (2002). Browsing support by the keyword extraction from a user's browsing history. *IEICE Technical Report, NLC. Natural Language Understanding and Models of Communication*, *101*(711), 85–92.

Goddard, C. (2002). The search for the shared semantic core of all languages. In C. Goddard, & A. Wierzbicka (Eds.), *Meaning and universal grammar - Theory and empirical findings* (Vol. 1, pp. 5–40). Amsterdam, Netherlands: John Benjamins. doi:10.1075/slcs.60.07god

Hiemstra, D., & de Jong, F. M. G. (2001). Statistical language models and information retrieval: Natural language processing really meets retrieval. *Glot International*, *5*(8), 288–293.

Jizba, R. (2007). Searching, part 4: Recall and precision: Key concepts for database searchers. *Searching: Introduction and General Topics*. Creighton University CDR. Retrieved November 23, 2013, from https://dspace.creighton.edu/xmlui/handle/10504/7292

Kawahara, D., & Kurohashi, S. (2013). *A Japanese morphological analysis system JUMAN*. Kurohashi Kawahara Laboratory. Retrieved November 8, 2013, from http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN

Kiselev, D., Rzepka, R., & Araki, K. (2013). Improving search query matching for electronic TV program guide data extraction. In *Proceedings of 2013 IEEE Seventh International Conference on Semantic Computing* (pp. 146-149).

Kumar, R. (2010). *Theory of automata, languages and computation*. Tata McGraw-Hill Education.

Kwak, M., Leroy, G., & Martinez, J. D. (2011). A pilot study of a predicate-based vector space model for a biomedical search engine. In *Proceedings of the 2011 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW)* (pp. 1001-1003). IEEE.

Lieber, R. (2004). *Morphology and Lexical Semantics, 104*. Cambridge University Press. doi:10.1017/CBO9780511486296

Lin, D., Church, K. W., Ji, H., Sekine, S., Yarowsky, D., Bergsma, S., & Narsale, S. (2010). *New tools for web-scale n-grams*. LREC.

Maruoka, A. (2011). *Concise guide to computation theory*. Springer. doi:10.1007/978-0-85729-535-4

McCandless, M., & Hatcher, E. (2010). *Lucene in action* (2nd ed.). Manning Publications.

Miller, E., Shen, D., Liu, J., & Nicholas, C. (2006). Performance and scalability of a large-scale n-gram based information retrieval system. *Journal of Digital Information*, *1*(5), 1–25.

Ptaszynski, M., Rzepka, R., Araki, K., & Momouchi, Y. (2012). Annotating syntactic information on 5.5 billion word corpus of Japanese blogs. In *Proceedings of the 18th Annual Meeting of The Association for Natural Language Processing (NLP-2012)* (pp. 385-388).

Sasaki, Y. (2007). The truth of the F-measure. *Teach Tutor Mater*, 1-5.

Smrz, P., & Schmidt, M. (2009). Information extraction in semantic wikis. *SemWiki, 464*.

Tanabe, T., Tomiura, Y., & Hitaka, T. (2000). Context free grammar expressing dependency constraint and its application to Japanese language. *Information Processing Society of Japan Journal*, *41*(1), 36–45.

Tummarello, G., Cyganiak, R., Catasta, M., Danielczyk, S., Delbru, R., & Decker, S. (2010). Sig. ma: Live views on the web of data. *Web Semantics: Science. Services and Agents on the World Wide Web*, *8*(4), 355–364. doi:10.1016/j.websem.2010.08.003

Wang, K., Thrasher, C., Viegas, E., Li, X., & Hsu, B. J. P. (2010). An overview of Microsoft Web N-gram corpus and applications. In *Proceedings of the NAACL HLT Demonstration Session, Association for Computational Linguistics* (pp. 45-48).

Wierzbicka, A. (1996). *Semantics: Primes and universals*. Oxford University Press.

Yamasaki, T., Manabe, T., & Kawamura, T. (2008). *Implementation of TV-program navigation system using a topic extraction agent. Computer Software* (pp. 41–51). Tokyo, Japan: Japan Society for Software Science and Technology.

ENDNOTES

[1] The URLs http://tv.yahoo.co.jp/ and http://www.tvguide.or.jp/ can be used to access two examples of such sites.

[2] Here and onward, examples in Japanese are followed by the romanization and translation. Romanizations are italicized. The underlined words (written in Japanese characters) are mandatorily matched by the system, other words may be ignored. For clarity, romanizations and translations of the mandatorily matched words are also underlined.

[3] http://www.google.com/intl/ja/insidesearch/howsearchworks/thestory/

[4] See http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?JUMAN for an English translation of a manual for one of JUMAN releases.

[5] Goo Dictionary at http://dictionary.goo.ne.jp/

[6] Sanseido Web Dictionary at http://www.sanseido.net/

[7] In this research the percentage refers to search queries in the English language only.

Denis Kiselev was born in Russia (formerly the USSR) in 1973. Received a diploma in Translation, Interpretation and Teaching from Nizhny Novgorod State Linguistic University, Russia in 1997. Received an M.A. degree in Computational Linguistics from Hokkaido University, Japan in 2012. Currently, is a doctoral candidate student at the Graduate School of Information Science and Technology, Hokkaido University, Japan. Interested in Information Retrieval, Morphological, Syntactic and Semantic Analyses.