Dictionary search based on the target word description

Slaven Bilac, Wataru Watanabe, Taiichi Hashimoto, Takenobu Tokunaga and Hozumi Tanaka
Department of Computer Science
Tokyo Institute of Technology
Tokyo, Japan
{sbilac,wataru,taichi,take,tanaka}@cl.cs.titech.ac.jp

Abstract

Users often experience difficulty lexicalizing the word they want to lookup or use in the text they are writing. In order to help such a user to access the desired word we propose a novel dictionary search method where the user can enter the description/definition of the target word rather than the word itself in order to look it up in the dictionary. Japanese input is first parsed by a morphological analyzer and then compared to the dictionary definitions to obtain the closest matching concept in the dictionary. Furthermore, the relations among concepts obtained from the concept dictionary are exploited to improve the system accuracy.

1 Introduction

Dictionaries in electronic format have become common place during the last ten years. Their advantages over paper dictionaries are numerous: fast, random access; ability to jump/navigate between several dictionaries; ability to adjust the displayed information to suit one's needs. Nonetheless, currently available dictionaries/dictionary interfaces fall short in one important aspect: accommodating the user with imperfect knowledge of the word he is trying to lookup. Most dictionaries support lookups only based on the correct, prescribed spelling of the word or support incomplete input based on crude string matching techniques such as regular expressions or shortest edit distance.

In many cases this is unsatisfactory since the user is unable to provide the correct information either because he cannot think of the word that he knows or because he does not know the word (as is often the case with the language learner). In cases like this, the dictionary search is unsuccessful and the users end up frustrated.

The goal of this research is to create a more intuitive, user-friendly dictionary search mechanism which will allow the user to look up the desired word without knowing its prescribed spelling. In this paper, we will describe how we go about helping the user to lookup the word that is on his mind but he was unable to lookup with conventional interfaces.

In Section 2 we describe the problem we are trying to address. Then we describe the implementation of the prototype system in Section 3. Finally, in Section 4 we evaluate the current implementation.

2 Problem of incorrect/incomplete input

With the correct input available, dictionary lookup is straightforward. However this is often not the case. Imagine a user who wants to find the word expressing the meaning "the food that cow chews over and over" but cannot think of word "cud." User cannot use conventional dictionary interfaces to use the information he knows about this word in order to look it up.[1]

This problem is different from the problem of incorrect or partially correct input of a known word which has been addressed in previous research (if only in limited fashion). For example, for French,

[1] However a simple search on Google gave the desired results in this case.

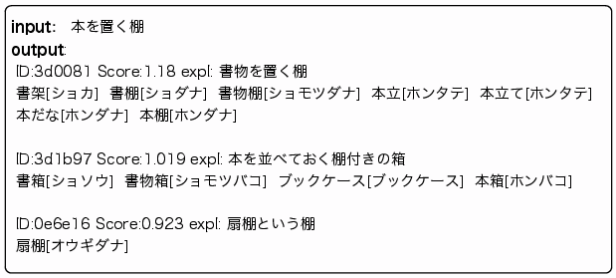

Figure 1: Example search

the system by Zock and Fournier (2001b) tries to account for confusion between the phonetic form of the word and its spelling. For Japanese, FOKS system (Bilac et al., 2003) allows lookup of words based on erroneous reading estimates of kanji characters contained in the word. Both of these systems explore mappings between characters and their pronunciation to account for inaccurate input.

Although it is feasible to handle the problem of access on the basis of an individual word as input (Zock, 2002; Zock and Fournier, 2001a), in this paper we address the search starting with a multi-word user input (El-Kahlout and Oflazer, 2004). The hypothesis is that even though the user does not know the word he wants to lookup, he can give a description of the word. Such a user would benefit from a dictionary allowing lookup based on the description he can provide. Figure 1 gives an example of the dictionary search.

3 Implementation

In order to allow the user access to the dictionary entries based on the description, we need to compare the input with the definitions from the concept dictionary (EDR, 1995). Since concepts are identified only with numerical codes, we use the concept definitions used by developers and human users to make it easier to understand what the concept represents. Translating this into IR vocabulary, the user input is the query and the concept definitions are the documents. Our goal is to find the set of the most relevant documents in response to the user query. Once the most relevant concepts are located, it is straightforward to obtain the dictionary entries which lexicalize them. We opted for using concept definitions rather than word definitions since concepts dictionary provides additional hierarchy information which can be used to improve similarity measures.

In the preparatory stages we parse all dictionary definitions with the ChaSen morphological analyzer (Matumoto et al., 2002)[2] and generate the frequency files necessary for GETA IR engine[3]. The frequency files reflect the term frequencies in each definition. We decided to use GETA engine since it allows for changing of the similarity measure used to evaluate which documents are relevant to the query.

3.1 Traditional similarity metrics

The starting point of our system are some standard similarity metrics used in IR (Tokunaga, 1999). We evaluate them separately and then augment them with additional information obtained from the concept dictionary (see below). Here t represent each term in a query q or a document d.

The first metric we used is $tf.idf$. Here the $tf.idf(t,d)$ is the product of the term frequency in a document $tf(t,d)$ and $idf(t,d)$, inverse document frequency weight calculated by Equation (1). In this equation N is the number of documents and $df(t)$ is a number of documents term t appears in.

$$idf(t) = \log \frac{N}{df(t)} + 1 \quad (1)$$

$$\overrightarrow{wq} = (tf(t_1,q), \cdots, tf(t_m,q)) \quad (2)$$

$$\overrightarrow{wd} = (tf.idf(t_1,d), \cdots, tf.idf(t_m,d)) \quad (3)$$

Then, we can rewrite the query and each document as vectors $\overrightarrow{wq}$ and $\overrightarrow{wd}$ (Equations 2 and 3) and calculate the similarity of two vectors as given in Equation (4). In this case the dot product of vectors is normalized by the sum of term frequencies in the document.

$$sim(q,d) = \frac{\overrightarrow{wq} \cdot \overrightarrow{wd}}{\sum_{n=1}^{m} tf(t_n,d)} \quad (4)$$

The second measure we used is cosine (cos). The individual vectors are calculated as above but the dot product is normalized by product of vector lengths as shown in Equation (5).

$$sim(q,d) = \frac{\overrightarrow{wq} \cdot \overrightarrow{wd}}{|\overrightarrow{wq}| \cdot |\overrightarrow{wd}|} \quad (5)$$

[2] http://chasen.aist-nara.ac.jp/
[3] http://geta.ex.nii.ac.jp/

The third measure tested is modified cosine (cos_m) (Matsuzaki et al., 1997). For this measure rather than using tf of each element in the query vector, the element is mapped to a binary value as given in Equation (6). The resulting vector (Equation 7) is then used to calculate the similarity as given in Equation (5).

$$a(t,q) = \begin{cases} 1 & \text{term } t \text{ is in the query} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

$$\overrightarrow{wq} = (a(t_1,q),\cdots,a(t_m,q)) \quad (7)$$

3.2 Using the concept hierarchy

As the next step we look at possible ways to use additional domain specific information to improve the performance. Since we can obtain concept relations from the EDR concept dictionary we use them to alter the similarity measures.

The parent concept definition usually contain more abstract terms relevant to the definition of the child concept. Thus, we expand dictionary definitions of a concept with the definitions of its parents according to Equation (8). Here the $tf_o(t,d)$ is the original term frequency in the document, $p(d)$ is the set of all parent definitions and $tf_n(t,d)$ is the new term frequency.[4] Based on obtained counts, the similarity measure par can be calculated as in Equation (5).

$$tf_n(t,d) = \alpha tf_o(t,d) + \beta \sum_{dp \in p(d)} tf_o(t,dp) \quad (8)$$

Second, we use the set of heuristics to extract the most significant term in the user input (Shotsu, 2003), convert it to a concept m and calculate a score for all adjacent concepts c (Equation (9)). Here $depth(c)$ is a depth of the concept in the hierarchy, and $MSCA(c,m)$ is the deepest ancestor node of both c and m. The calculated score is then combined with the GETA score to obtain the new value mc as in Equation (10).

$$Re(c,m) = \frac{2 \times depth(MSCA(c,m))}{depth(c) + depth(m)} \quad (9)$$

$$n\_score(c) = \left(1 + \frac{Re(c,m)}{\gamma}\right) \times score(c) \quad (10)$$

[4]Weights α and β adjust the influence of terms in parent concept definition.

Third, we directly check whether the user input matches one of the concept definitions. If that is the case, we return the concept directly. This heuristic is labeled syn during the experiment phase.

4 Evaluation

The biggest problem in evaluating the proposed system is the need for a collection of freely occurring dictionary queries similar to the queries that the system is trying to accommodate. Since we were unable to locate such a collection we resorted to using dictionary definitions from a different dictionary.

We randomly extracted 466 dictionary entries from the Iwanami Japanese dictionary (Nishio et al., 1994) and used the gloss of each entry as the query and the entry itself as the desired result. A complete set of 400,000 EDR concepts was then searched with each query as input and all words which lexicalize the relevant concepts as output. Since there were some discrepancies in punctuation and formatting between the two dictionaries, we removed all punctuation from the input and considered correct all answers which differed from the target word only in presence of *suru* verbal ending or in the *okurigana* ending.

The results of the experiment are given in Table 1 for the three standard measures (i.e. $tf.idf$, cos and cos_m) as well as for the three measures/heuristics using the EDR concept dictionary (i.e. par, mc and syn). Note that the three latter measures use cos_m as the base similarity calculation method.[5] From Table 1 we can see that cos_m yields the best results in absence of concept hierarchy information. However, supplementing the base similarity measures with domain specific information results in slight improvements. For the top-30 case, we can see that expanding the definition of a concept with that of its parents results in 2.1% increase whereas the combination of the three results in a 4.5% increase in coverage. Nonetheless, the highest accuracy rate of 52.8%, although high by IR standards, leaves room for improvement. Furthermore, in about 110 cases, the correct word was not returned among the top 1000 results. This is mostly because the definition sentence was too short to make a connection with the

[5]Due to time constraints we did not evaluate the influence of each measure using the concept dictionary separately, but only in compound form.

TOP-n	$tf.idf$	cos	cos_m	par	$par + mc$	$par + mc + syn$
1	31 (06.7%)	94 (20.2%)	98 (21.0%)	105 (22.5%)	107 (23.0%)	**113 (24.2%)**
5	109 (23.4%)	154 (33.0%)	166 (35.6%)	169 (36.3%)	172 (36.9%)	**180 (38.6%)**
10	145 (31.1%)	182 (39.1%)	200 (42.9%)	200 (42.9%)	203 (43.6%)	**212 (45.5%)**
30	196 (42.1%)	216 (46.4%)	225 (48.3%)	235 (50.4%)	238 (51.1%)	**246 (52.8%)**
50	223 (47.9%)	229 (49.1%)	249 (53.4%)	256 (54.9%)	264 (56.7%)	**272 (58.4%)**
100	251 (53.9%)	248 (53.2%)	272 (58.4%)	284 (60.9%)	288 (61.8%)	**296 (63.5%)**

Table 1: Comparison of different similarity measures

input sentence. This exemplifies the need to further explore the possibility of definition expansion with more detailed definitions/descriptions from a different source. Another possible method of improving the system is to reduce the search space (e.g. searching only for concepts in deeper levels of hierarchy). Furthermore, a significant problem for the system was the wide variation in spelling (e.g. use of different script for the same word) which reduces the accuracy significantly.

In the future we hope to implement a graphical interface to the system and extend it with additional navigational tools to enable access even in cases where similarity metrics failed to yield the desired result. Only when such extensions are available will the user be able to take full advantage of the system.

5 Conclusion

It is a common case that the user cannot provide canonically correct input when searching the dictionary. Hence there is a need to create a more robust search mechanism which allows lookup based on partial or erroneous input. In this paper we describe a system allowing lookup of dictionary entries based on the description of the target entry. User input is parsed and then compared with definitions contained in the dictionary using a variety of similarity metrics. In the preliminary experiments more than 50% of desired words were contained in the top 30 candidates returned by the system.

References

S. Bilac, T. Baldwin, and H. Tanaka. 2003. Improving dictionary accessibility by maximizing use of available knowledge. *Traitement automatique des langues*, 44:2.

EDR. 1995. *EDR Electronic Dictionary Technical Guide*. Japan Electronic Dictionary Research Institute, Ltd. (In Japanese).

I. D. El-Kahlout and K. Oflazer. 2004. Use of Wordnet for retrieving words from their meanings. In *Proc. of the Global Wordnet Conference (GWC2004)*, pages 118–123.

T. Matsuzaki, T. Miura, Y. Komata, T. Saitou, K. Yamada, and H. Nakagawa T. Mori. 1997. Contents retrieval system of japanese manual. *Information Processing Society of Japan SIG Notes*, 97-NL-117:113–120.

Y. Matumoto, A. Kitauchi, T. Yamashita, Y. Hirano, H. Matsuda, K. Takaoka, and M. Asahara. 2002. Morphological analysis system ChaSen version 2.2.9 manual.

M. Nishio, E. Iwabuchi, and S. Suitani, editors. 1994. *Iwanami Kokugo Jiten*. Iwanami Shoten, 5th edition. (in Japanese).

Y. Shotsu. 2003. Kokugo yiten to shisoorasu no tougou ni kan suru kenkyuu. Master's thesis, Tokyo Institute of Technology. (in Japanese).

T. Tokunaga. 1999. *Jouhou kensaku to gengo syori*. University of Tokyo Press. (in Japanese).

M. Zock and J.-P. Fournier. 2001a. How can computers help the writer/speaker experiencing the tip-of-the-tongue problem ? In *Proc. of RANLP*, pages 300–302.

M. Zock and J.P. Fournier. 2001b. Proposal for a customizable, psycholinguistically motivated dictionary to enhance word access. In *Proc. of VII Simposio Internacional de Comunicacion Social*, pages 410–413.

M. Zock. 2002. Sorry, what was your name again, or how to overcome the tip-of-the tongue problem with the help of a computer? In *Proc. of the SemaNet workshop COLING2002*.