Towards a user-friendly dictionary interface

Slaven Bilac and **Michael Zock**
Department of Computer Science
Tokyo Institute of Technology
Tokyo, Japan
sbilac@cl.cs.titech.ac.jp,zock@limsi.fr

Abstract

The ultimate goal of a dictionary is to reveal the information it contains and to give the user what he is looking for (word, definition, grammatical information). While looking straight-forward, this is not an easy task to define, as users have different needs, expectations and knowledge, all of which influence the way of using this resource. In this paper we discuss how computers can help people to find wanted information in an efficient and intuitive way. Depending on the user's goal, encoding (speaking/writing), decoding (listening/reading), or both (translation), he will have different needs with regard to the information contained in the dictionary. We will present some methods and guidelines to build a dictionary interface that meets (or addresses) some of these needs.

1 Introduction

Electronic dictionaries have become increasingly popular in recent years. Nonetheless, current dictionary interfaces generally fail to fully take advantage of the many possibilities computers offer (flexibility of input, interactive, multi-criteria-based searching). This is probably due to the fact that Electronic Dictionaries are basically only electronic versions of their paper counterparts. Unfortunately, the change of the medium has not been followed with a corresponding change of the facilities provided for the dictionary user. Actually, these "new" dictionaries obey, most of the time, the conventions of their paper ancestors, be it with regard to content, size, formating or alphabetic ordering. Obviously, all this puts severe constraints on access, be it for words or their conceptual counterpart (meaning). Including more entries and examples in the dictionary requires not only huge investment in terms of editorial man power, but also raises various problems concerning example appropriateness and usefulness. We consider these issues beyond the scope of this paper. What we are really interested in is to discuss the problem of ordering the data, i.e. the internal/external organisation of the dictionary, as this will have a direct bearing on information access. What is a dictionary good for if one cannot access the data it contains?

Obviously, the dictionary user will benefit from the richness of the resource only if he is able to quickly find the needed information (target word, meaning, etc.). Of course, lookup is quite trivial when the user knows the correct spelling (orthographic presentation) of the target word. Unfortunately, this is not always the case, hence additional tactics are necessary to enable access of the desired information. Let us elaborate on this idea a bit more.

A dictionary user typically pursues one of two goals (Humble, 2001):

a) as a decoder (reading, listening), he may

want to find the definition or translation of a specific target word, while

b) as an encoder (speaker, writer) he may want to find a word or expression that expresses well not only a given concept, but is also appropriate in a given context.

Consequently, readers and writers come to the dictionary with different expectations. There are clear differences both in terms of what they expect as an output and what they can provide as input (knowledge available at the onset). The decoder generally knows the word he is interested in, but he may not know its meaning. Yet, having a clear idea concerning the form of the word to lookup does not guarantee that one is able to do so. For example, a user reading some Japanese text may want to lookup the translation of the word 頭上 *zujou* "above,overhead". Yet, in order to do so, he would have to be able to tell the dictionary what specific word he is looking for, that is, he would need to write (input) the corresponding form into the dictionary interface designed for that purpose. This supposes not only that he knows how to write in Japanese (kana, kanji), but also that he is able to read the word as *zujou*. A user may also be unable to lookup an English word like */doʊ/*[1], heard in a conversation. The problem here are his imperfect spelling capacities rather than his incapacity to read the word. In cases like these users end up being frustrated, as they cannot accomplish their goal.

A user trying to encode his thoughts has a different problem to deal with. He generally knows the meaning, or at least part of the idea he wants to convey, but he cannot access the word in time. Suppose, the user were looking for a word expressing the following ideas : "domesticated animal, giving milk suitable for making cheese". Suppose further that he knew that the target word was not *cow*. Of course, none of this is sufficient to guarantee the access of the intended word *sheep*. Consequently, he would neither be able to use (input, write) this word in the text he is about to produce, nor be able to look up its definition. Again, as a result, the user would fail to achieve his goal.

Clearly, a user facing such problems can experience frustration, and in the case of a language learner (where unknown words, and problems of rapid word access are common) this could yield serious temporal and motivational setbacks.

In this paper we will consider what electronic dictionaries and dictionary interfaces can do to improve the situation, in particular with regard to information access. The paper is structured in the following way: we will start by reviewing currently available systems (Section 2). In Section 3 we will outline a framework for building an improved dictionary interface, the goal being to help the user to find rapidly the wanted information (target word). This guidance is sensitive not only to the knowledge the user has at the onset but also to his goal (encoding vs. decoding). Finally, in Sections 4 and 5 we provide specific solutions to the problem of word access, outlining the current state of implementation and remaining issues.

2 Existing Dictionaries/Interfaces

2.1 Form-based access

In paper dictionaries it is very common to order all entries alphabetically, expecting the user to search the dictionary in that very same order. Due to physical constraints it was not possible to provide several access methods. Notable exception to the rule are thesauri and kanji dictionaries which commonly provide several indexes, enabling the user to search for the desired entry in several ways. For example, Halpern (1998) allows for six different ways to search for an entry. Each of these schemes requires correct knowledge of a given search criterion to enable the lookup of the desired entry.[2] We will refer to this as exact-match search. In cases where the input criteria is the orthographic representation of the word exact-match search is supplemented with substring-, prefix- or suffix search, whereby the dictionary entries are matched with

[1] Here the pronunciation of *dough* is given in IPA.

[2] Here again, the notable exception are kanji dictionaries which commonly list characters under two different radical or stroke number indexes.

the user input on a partial basis. More sophisticated implementations allow for the use of regular expressions (REGEXP) in specifying the search criteria, hence improving the overall chances to access the desired information in the dictionary. For example, a user looking for the word *consequence* can access this entry by inputing *(c|k).+nce*[3] into the dictionary interface. We will refer to these methods as REGEXP-based search methods. While these methods are highly effective if the user is able to correctly specify the desired search criteria, they provide little help if he is not able to do so. Furthermore, REGEXP-based search methods work only for dictionaries employing alphabetic ordering. For example, kanji dictionaries are generally indexed in terms of kanji-radicals and numbers of strokes. Yet, these criteria cannot be represented in a REGEXP format such as to allow matching of the desired entry. In addition, REGEXP-based search criteria are often not very satisfying due to their large number of hits (possible candidates), which reduces their usefulness considerably for general cases.

Various reading-aid systems take advantage of the computers' information processing capacities, providing morphological and syntactic analysis of texts at the sentence-level. The Reading Tutor[4] (Kawamura, 2000; Kitamura and Kawamura, 2000) and Asunaro[5] are examples of such systems for Japanese. (Nishina et al., 2000; Nishina et al., 2002). Once the sentence is parsed, each word is converted into a dictionary form which is then looked up in the dictionary, thus removing this responsibility from the user. Additional information, such as semantic description or syntactic trees, are displayed to improve the sentence-level understanding. The major shortcoming of such a system is that the text (or at least a single complete sentence) needs to be converted into electronic form before a system can be used. Similar systems exist for other languages. For example, MoBiDic provides morphologic analysis and dictionary-lookup for Hungarian (Proszeky, 1998) , whereas Papillon[6] provides lemmatization for French. Such facilities are very efficient and certainly very useful for accurate input, but they generally fail to help the user when the input is not perfect.

Of course, the problem of erroneous input has been addressed by systems containing a spell-checker. Among more recent works are the spell-checkers based on the erroneous channel model, allowing for multi-character errors. While the initial implementation by Brill and Moore (2000) was based only on orthographic information, follow-up work by Toutanova and Moore (2002) improved the performance by adding phonetic information (a major source of confusion). The problem with spell-checking systems is that they normally apply only to words absent from the system dictionary. Hence they are of little help in such (common) cases where the target word is substituted by a correctly spelled, but semantically different word (e.g. *whether* instead of *weather*).

2.2 Meaning-based access

Attempts to improve word access on the basis of meaning, have been far less numerous. The best known are Roget's thesaurus (Kipfer, 2001) and WordNet[7] (Miller et al., 1993). Roget's thesaurus has been developed to enable word access on the basis of topical relatedness (game of tennis: *umpire*, *racket*, *player*, *ball*, etc.). On the other hand, in a psycho-linguistically motivated dictionary like WordNet access can be performed via a limited set of semantic links between synonym sets (synsets). The semantic relations are organized into an inheritance network such as to allow for simple navigation. WordNet dictionaries have been developed for several Indo-European languages[8] (Vossen, 1998). While this work is clearly important, as it seems to correspond to the way information is organized in our mind (Baddeley, 1982), there is nevertheless room for improvement. For ex-

[3] Here *(c|k)* matches either *c* or *k* and .+ matches one or more alphabetic characters. Other characters match themselves.

[4] `http://language.tiu.ac.jp/`

[5] `http://hinoki.ryu.titech.ac.jp/`

[6] `http://www.papillon-dictionary.org`

[7] `http://www.cogsci.princeton.edu/~wn/`

[8] `http://www.illc.uva.nl/EuroWordNet/`

ample, the number of semantic relationship currently accounted for is too limited: people store many more associations, especially at the syntagmatic level, than WordNet has. There are relationships, easily perceived by humans, but clearly beyond the scope of WordNet. This is the case for a cause-effect relationship like *injury* and *pain*. Also beyond its scope are the above-mentioned topical relationships such as *tennis player*, *racket* and *ball* (the authors of WordNet are aware of this shortcoming and refer to it as the "tennis problem" (Fellbaum, 1998)). Finally, limited research has been conducted aiming to supplement conventional dictionary access by use of relationships available in WordNet (Chanier and Selva, 1998).

3 What can computers do to help?

Above we have introduced several systems aiming to help the user locate the desired information. Nonetheless, several problems have not been addressed yet. The basic goal of a dictionary interface is to provide an intuitive, coherent and effective way for accessing the desired information contained in the database. When a user is able to provide the canonical, orthographic representation of the target word, the problem is trivial. Direct-matching methods will suffice to locate and display the desired entry. However, users are often unable to provide such perfect input. Either they lack knowledge of certain aspects of the language (as is often case with foreign learners), or they are not able to retrieve the stored knowledge of the word (e.g. the word's base form, lemma), although they know the word and its canonical form.

Hence, additional bridges need to be built to shift the burden from the user to the dictionary interface, such as to guide him (possibly interactively) to the target (word, definition, etc.). To create such a dictionary interface, we have to accomplish two goals:

a) Provide the user with an interface that allows him to tell the system what information he has concerning the word he is looking for (reading, number of characters/syllables, related words, meaning).

b) Provide means to allow navigation in a natural way (the way people do), that is by following the links connecting words in a huge associative network.

In the following sections we will describe two separate interfaces, aiming to achieve these goals. One focuses on decoding, while the other one deals with encoding (expression). Currently, the interfaces are separate, but they could well be merged into a unified whole.

4 Helping the user with a decoding goal

Written Japanese makes heavy use of kanji (ideograms), which makes dictionary lookup very difficult for a non-native speaker, even if he can see the character in the text.[9] However, even when the prescriptive reading of the whole word is not available, the user often possesses some knowledge of readings of the characters forming the word. Provided the user can input this partial knowledge into the system interface, the system can calculate what dictionary entry the user might be interested in, and thus guide him to the target entry. For example, a user not familiar with the prescriptive reading *zujou* of 頭上 *zujou* "overhead" might nevertheless be able to estimate a reading *toujou*, since he knows a more common reading for 頭 (i.e. *tou*). Given the input reading *toujou* and the knowledge of potential readings of 頭 (i.e. *tou* and *zu*), the system can bridge the gap between the input reading and the target entry 頭上.

The FOKS[10] (Forgiving Online Kanji Search) system is based on this kind of intuition and on the knowledge (analysis) of the phonetically-polymorphic properties of kanji characters. It allows the user to lookup a kanji word without necessarily knowing its correct reading (Bilac et al., 2002). The current implementation handles errors due to reading errors (incorrect reading of some, or all of the component characters of the string), inappropriate application of

[9] Note that minimum edit distance methods are not applicable to kanji due to the large amount of information contained in each character, yielding vastly different results with each edit.

[10] http://www.foks.info/

phonological and morphological rules pertaining to string formation and vowel/consonant length confusion. The error handling mechanism does not use REGEXP, as this method cannot deal with all common error types (Baldwin et al., 2002) neither can it discriminate between all alternate readings.[11] Instead, the system is based on a generative probabilistic model, computing the word's overall reading by concatenating the readings of the words' individual characters and by subsequently applying morpho-phonological rules. For example, given a string like 頭上 the system would calculate the probability of each character taking a reading (e.g. *tou*, *zu* and *atama* readings for 頭 and *jou*, *ue* and *agaru* readings for 上) and then compute the probability of the overall reading undergoing a change (e.g. *toujou* becoming *toujo* or *doujou*[12]).

Given some user-input the system does not make any assumptions concerning its quality (correct or incorrect), rather it suggests all plausible dictionary entries connected to the input, letting the user choose the most appropriate entry from a list of candidates. Hence the system encourages the user to apply the available knowledge, even in cases where this knowledge is not complete/perfect. Note that in the above case, conventional dictionaries would not yield the desired result, therefore failing to help the user to apply the knowledge he might have concerning a target word.

4.1 Necessary extensions to FOKS

While FOKS does provide systematic and statistically sound error handling in dictionary lookups based on the reading, it fails to allow the user to provide some of the information available about the target word. Since the user can see the word, he can often provide additional information (such as, number of characters of the word, number of strokes, or meaningful subparts of some characters, etc.), to further constrain the search process, thus limiting the number of potential candidates. Such criteria can currently be used for searching kanji dictionaries, but their integration with word-based access methods would further increase the ability to guide the user towards the target word.

It should be noted that, although the current implementation of the system deals only with written Japanese, the same approach could be useful in an oral setting (conversation). Indeed, a listener may well be unable to type into the dictionary interface the word heard in a conversation, because of the discrepancy between the phonetic form (sound) and the spelling of the word, i.e. orthographic form stored in the dictionary (Zock and Fournier, 2001).

5 Helping the user with an encoding goal

The method described above is useful for someone who wishes to translate a word he can see, but cannot read/write. Yet, quite often, the user will not have anything visible to start out with. Instead he will have a definition (or, part of it), a related word (synonym, hyperonym), etc. In such case we can try to guide the user by allowing him to get to the desired target word from this underspecified, incomplete input. To this end we can use lexical- (associations), conceptual- (definition, relation to other concepts) and metalinguistic (number of syllables) information to get closer to the target word: every time the user provides information the system will display a set of candidates (list of dictionary entries) from which the user can choose. Obviously, the more specific the input the smaller the search space. By repeating this process several times, the user should get closer to the target word, ultimately finding the word he is looking for.

5.1 Access by meaning: navigation in a huge associative network

As mentioned above, research shows that people don't store words in alphabetic order (Deese, 1965; Baddeley, 1982; Aitchin-

[11] As mentioned by one of the reviewers, Jeffrey's Server (`http://linear.mv.com/cgi-bin/j-e/jis/dict/`) handles cases of vowel/consonant length confusion. However, this system is unable to handle substitutions of character readings so searches like *toujou* would not lead to the entry 頭上.

[12] Vowel shortening/lengthening and voicing exemplified here respectively are common causes of reading errors in learners.

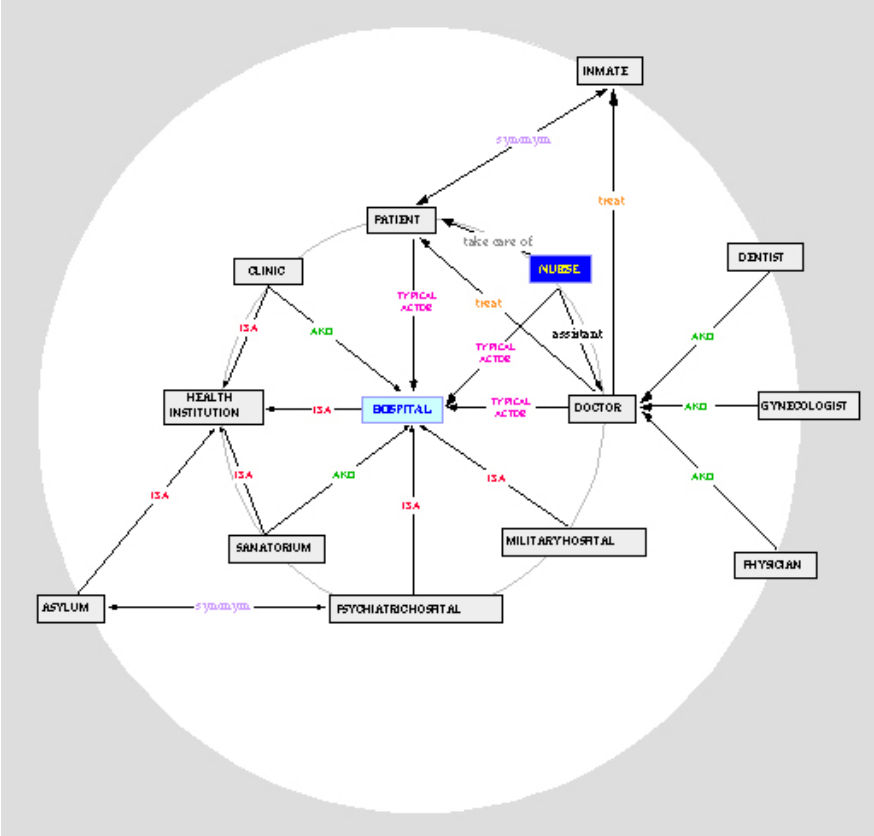

Figure 1: A section of semantic network stored in system memory (internal representation)

son, 1987), rather they store words by meaning, by relations (links/associations between words/concepts) and by form (phonemic representation). As the speaker normally starts from a meaning representation (concept, message) we aim to use THIS as the starting point, helping him to find the appropriate word by navigating in a dynamically built network. In other words, we start from the assumption that the mental dictionary is a huge semantic network composed of words (nodes) and associations (links), either being able to activate the other. Finding a word amounts thus to entering the network at a certain point and following the links leading to the target word. While being unable to "produce" the desired word, a speaker being in the tip-of-the-tongue state (TOT-state) is still able to "recognize" it in a list. If the list does not contain the exact word, he is generally able to decide which of the words leads in the right direction, i.e. which word is most closely connected to the target word.

Suppose you wanted to find the word *nurse* (target word), yet the only token coming to your mind were *hospital*. Since the system internally stores link information, and since all words are interconnected, any word coming to your mind has the potential to evoke all other directly or indirectly related words. Put differently, the target word is likely to be a part of a semantic network, with the remembered word (source word), e.g. *hospital*, in the center, and the associated words as immediate satellites. Figure 1 gives only an abstract, internal representation, whereas Figure 2 shows the display a user might see. The difference between these two forms of representation is rooted in readability and navigability. Figure 1 is a lexical view, which represents well the fact that words are highly interconnected, but it blurs the idea of class membership, as this information is distributed. Yet, words' class membership is vital

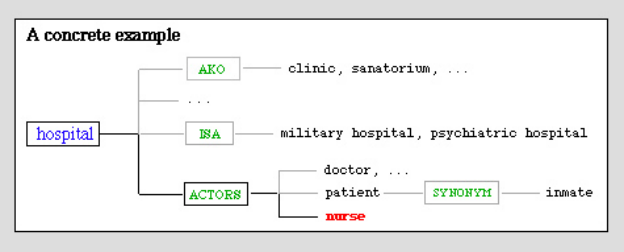

Figure 2: Navigational aid presented to the user

for quick navigation. This is why we use the relational view representation as the actual user interface. As seen in Figure 2, words/links are presented in clusters. Each word cluster corresponds to a specific link. The assumption is that the user will use this information in order to jump quickly from one group to the next expediating the search: "`a kind of`" (*clinic*, *sanatorium*); "`examples`" (*military hospital*, *psychiatric hospital*); "`typical actors`" (*doctor*, *patient*, *nurse*). Obviously, if one is looking for a word like *nurse*, one knows that there are more chances to find it under the heading "`typical actors`", than the heading "`a kind of`".

5.2 Implementing the idea of navigating in a huge semantic-network

We have tried to show that, in order to support an encoder being in the TOT-state, one needs to do two things: add to an existing electronic dictionary information that people tend to associate with a word (enriching the semantic network), and provide a tool to navigate in the network. That is, we have raised and partially answered the question of what kind of information semantic networks need to have in order to be able to help an encoder being in TOT-state. Actually our basic proposal is to build a system akin to WordNet, but containing many more links (in particular on the horizontal plane i.e. syntagmatic relations). These links are associations, whose role consists in helping the encoder to find either ideas (concept/idea/word) related to a given stimulus (brainstorming), or to find the word he is thinking of (word access).

One approach to identifying the most useful associations, is by considering relevant work in linguistics and collecting data by running psycholinguistic experiments. For example, one could identify search patterns,[13] ask people to label the links for the words (associations) they have given in response to a stimulus (word); or one could also ask them to make explicit the nature of the links between word couples (e.g. *apple-fruit*, *lemon-yellow*, etc.). While this method might be superior in the quality of the obtained data, it requires extensive man–power and time thus making the implementation difficult.

The other approach would be to look at extracting the necessary information automatically. The basic idea is to extract co-occurrence and collocation data from the corpus (Rapp and Wettler, 1991; Wettler and Rapp, 1992; Ferret, 2002) and to use ontologies to qualify the links between words (or concepts) (Agirre et al., 2000; Stevenson, 2002). While this approach might work fine for couples like *coffee-strong*, or *wine-red* (since an ontology would reveal that red is a kind of "`color`", which is precisely the link type; i.e. association), it is not clear it could reveal the nature of the link between *smoke* and *fire*, which most humans would immediately recognize as a causal link. This is only one of the few aspects that current state-of-the-art-technology cannot address when automatically extending existing lexical resources. However, we are confident that we can make solid strides towards intuitive and easy dictionary access by employing similar technologies.

6 Conclusion

In this paper we have discussed some of the shortcomings of currently available electronic dictionaries. Our focus has been on access. We presented several improvements by taking into account the user's goal: encoding or decoding. While the solutions offered have been partially implemented separately for the two tasks, we plan to merge them to provide a unified framework for dictionary lookup that is sensitive to both goals.

[13] One such pattern could be: give me the word of a bird with a long mouth and yellow feet that can swim.

References

E. Agirre, E. Hovy O. Ansa, and D. Martinez. 2000. Enriching very large ontologies using the WWW. In *Proc. of ECAI Ontology Learning Workshop*.

J. Aitchinson. 1987. *Words in the Mind: an Introduction to the Mental Lexicon*. Blackwell.

A. Baddeley. 1982. *Your memory: A user's guide*. Penguin.

T. Baldwin, S. Bilac, R. Okumura, T. Tokunaga, and H. Tanaka. 2002. Enhanced Japanese electronic dictionary look-up. In *Proc. of LREC*, pages 979–985.

S. Bilac, T. Baldwin, and H. Tanaka. 2002. Bringing the dictionary to the user: the FOKS system. In *Proc. of the 19th International Conference on Computational Linguistics (COLING 2002)*.

E. Brill and R. C. Moore. 2000. An improved error model for noisy channel spelling correction. In *Proceedings of the 38th Annual Meeting of the Association for Computarional Linguistics (ACL 2000)*, pages 393–399, Tokyo, Japan.

T. Chanier and T Selva. 1998. The ALEXIA system: the use of visual representations to enhance vocabulary learning. *Computer Assisted Language Learning*, 11:489–521.

J. Deese. 1965. *The structure of associations in language and thought*. Johns Hopkins Press.

C. Fellbaum. 1998. *WordNet: An Electronic Lexical Database and some of its Applications*. MIT Press.

O. Ferret. 2002. Using collocations for topic segmentation and link detection. In *Proc. of the 19th International Conference on Computational Linguistics (COLING 2002)*, pages 261–266.

J. Halpern, editor. 1998. *New Japanese-English Character Dictionary*. Kenkyusha Limited, 6th edition.

Ph. Humble. 2001. *Dictionaries and Language Learners*. Haag + Herchen.

Y. Kawamura. 2000. The role of the dictionary tools in a Japanese language reading tutorial system. Ljubljana University International Seminar. (In Japanese).

B. A. Kipfer, editor. 2001. *Roget International Thesaurus Indexed Edition*. HarperCollins, 6th edition.

T. Kitamura and Y. Kawamura. 2000. Improving the dictionary display in a reading support system. In *International Symposium of Japanese Language Education*. (In Japanese).

G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and Katherine Miller, editors. 1993. *Introduction to WordNet: An On-line Lexical Database*. Cognitive Science Laboratory, Princeton University.

K. Nishina, M. Okumura, S. Sugimoto, Y. Yagi, T. Abekawa, N. Totsugi, and F. Ryang. 2000. Development research on multilingual Japanese reading aid for foreign students with scientific background. *Research Report of Telecommunications Advancement Foundation*, 15:151–159. (In Japanese).

K. Nishina, M. Okumura, Y. Yagi, N. Totsugi, F. Ryang, S. Sugimoto, and T. Abekawa. 2002. Development of Japanese reading aid with a multilingual interface and syntax tree analysis. In *Proc. of the Eight Annual Meeting of The Association for Natural Language Processing (NLP2002)*, pages 228–231. (In Japanese).

Gabor Proszeky. 1998. An intelligent multidictionary environment. In *Proc. of the 17th International Conference on Computational Linguistics (COLING 1998)*, pages 1067–1071.

R. Rapp and M. Wettler. 1991. A connectionist simulation of word associations. In *Proc. of Joint Conference on Neural Network*.

M. Stevenson. 2002. Augmenting lexical similarity metrics. In *Proc. of the 19th International Conference on Computational Linguistics (COLING 2002)*, pages 953–959.

K. Toutanova and R. C. Moore. 2002. Pronunciation modeling for improved spelling correction. In *Proc. of the 40th Annual Meeting of the Associaiton for Computational Linguistics (ACL)*, pages 144–151.

P. Vossen, editor. 1998. *Euro WordNet: A Multilingual Database with Lexical Semantic Networks*. Kluwer Academic Publishers.

M. Wettler and R. Rapp. 1992. Computation of word associations based on the co-occurences of words in large corpora. In *Proc. of the Workshop on Very Large Corpora: Academic and Industrial Perspectives*, pages 84–93.

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