Storage does not Guarantee Access: The Problem of Organizing and Accessing Words in a Speaker’s Lexicon

Michael Zock\textsuperscript{1} and Didier Schwab\textsuperscript{2}

\textsuperscript{1}LIF-CNRS, 163, Avenue de Luminy, 13288 Marseille, France
michael.zock@lif.univ-mrs.fr

\textsuperscript{2}LIG-GETALP, University of Grenoble, 38041 Grenoble cedex 9
didier.schwab@imag.fr

Natural language production requires both a grammar and a lexicon. In this article, we deal only with the latter, trying to enhance an existing electronic resource to allow for search via navigation in a huge associative network. Our primary focus is on the structure of the lexicon (i.e. its indexing scheme). This issue has often been overlooked, yet it is crucial, as it determines to a large extent the chances of finding the word a language user (speaker/writer) is looking for. While researchers working on natural language generation (NLG) have given a lot of thought to \textit{lexicalization} (i.e. the mapping of meanings to forms), \textit{lexical access} has received no attention at all. Lexicalization is generally considered to be only a choice problem, the assumption being that stored data can always be accessed. While this may hold for machines, it does not always hold for people, as is well attested by the “tip-of-the-tongue” problem. A speaker may know a word, yet still be unable to access it. However, even machines may experience access problems. We illustrate this last point via a small experiment, showing how a well-known lexical resource (\textit{WordNet}) may fail to reveal information (words) it contains. Additionally, in this article we show how a lexicon might be organized or indexed to allow language users to find the words they are looking for quickly and naturally.

Key words: language production, lexical access, word finding, searching, spreading activation, navigation, association-based index

©2011 Institute for Cognitive Science, Seoul National University
1. Problem: finding the needle in a haystack

One of the most vexing problems in speaking or writing is the inability to recall or access a known word when needed. This kind of search failure, known as *dysnomia* or *tip-of-the-tongue* problem, occurs not only in communication, but also in other situations in which one is momentarily unable to recall something such as a date, phone number, past event, or a person’s name.

The problem with which we are concerned in this article is the problem of how to find words in the places where they are stored: the human brain or an external resource such as a dictionary. Our work being confined to lexical access, we have started to develop a semantic map and a compass to help language producers find the words they are looking for. More precisely, we try to build an index and a navigational tool allowing people to access words, no matter how incomplete their conceptual input may be. Our approach is based on psychological findings concerning the mental lexicon (Aitchison, 2003; Roelofs, 2004), that is, storage (representation and organization) and access of information in the human mind (Roelofs, 1992; Levelt et al., 1999), observed search strategies, and typical navigational behavior (Atkins, 1998; Thumb 2004). Before explaining the rationale underlying the construction of the dictionary of the language producer¹ we would like to challenge the following two assumptions: (a) that lexicalization is basically (only) a choice problem, and (b) that stored data (e.g., words) can always be accessed.

1.1 The author’s problem: choosing words, finding them, or both?

Given the expressive power of language,² natural language generation

---

¹ To avoid possible misunderstandings, our goal is not so much to build yet another lexicon as to add to it one or several indexes in order to ease navigation, as this is precisely what is lacking in most resources.

² Most ideas or concepts can be expressed via a set of means, i.e. semantically equivalent linguistic forms: *synonyms* (big-tall), *paraphrases* (minimize-limit the amount), *periphrases* (smarter-more smart), *circumlocutions* (scissor-tool for cutting paper), etc.
Memorisation of Words does not Guarantee Their Access

NLG researchers view lexicalization mainly as a choice problem (McDonnell, 1980; Stede, 1995; Reiter 1991). Though this may be correct, it does not follow that word access is not a problem worthy of consideration. Obviously, before making any choice, one must have something to choose from. Put differently, one must have accessed at least one word, if not more. Yet, since access does not seem to be a problem for machines—computers generally manage to retrieve stored information—NLG researchers consider lexical access as irrelevant. Hence, they do not address this issue at all.

While this attitude may be justifiable in the case of (fully automated) text generation, it is not acceptable in the case of language production by people, or interactive, computer-mediated language production (our problem here). Words may elude speakers at any moment, typically when they need them most, at the very moment of speaking or writing. Alas, the fact that speakers have memorized the words they are searching for does nothing other than to create frustration, as they feel deprived of something they know but cannot get hold of. This is generally the moment at which speakers reach for a dictionary, provided that they care and have the time and access to an appropriate resource.

While there are many kinds of dictionaries, very few of them are really

---

3 It is interesting to note that while lexicalization has been a much studied problem in NLG, —for excellent surveys see (Robin, 1990; Stede, 1995; Wanner, 1996), or, for some earlier work (Goldman, 1975), — the issue of lexical access has not been addressed at all, neither by any of the aforementioned authors nor by Cahill & Reape, 1999 or Ward (1988), who presents a problem catalog.

Conversely, there is an enormous amount of research in psycholinguistics published either as edited volumes (Bonin, 2004; Levelt, 1992; Marslen-Wilson, 1979), as monographs (Aitchison, 2003; Jarema et al., 2002; Stemberger, 1985), or as journal papers. Many of them are devoted to speech errors (Butterworth, 1982; Cutler, 1982; Fay and Cutler, 1977; Fromkin, 1973), the tip-of-the-tongue phenomenon (Brown and McNeill, 1996, Burke et al. 1991), or the lexeme/lemma distinction (Kempen and Huijbers, 1983; Roelofs, 1992). While all these publications deal with the problem of lexical access, they do not consider how computers can be utilized to help humans in their task. This is precisely our goal.

4 This experience is very similar to the feeling one has when looking for an existing object in the household or when trying to recall someone’s name. In all these cases something has been stored, but cannot be accessed in time.
helpful for the writer or speaker. To be fair though, one must admit that
great efforts have been made to improve the situation. In fact, there are
quite a few onomasiological dictionaries. For example, *Roget’s Thesau-
rus* (Roget, 1852), analogical dictionaries (Boissière, 1862, Robert *et al*.,
1993), *Longman’s Language Activator* (Summers, 1993) various network-
based dictionaries: *WordNet* (Fellbaum, 1998; Miller *et al*., 1990), *MindNet*
(Richardson *et al*., 1998), *HowNet* (Dong and Dong, 2006), *Pathfinder*
(Schvaneveldt, 1989) and (Fontenelle, 1997). Other proposals have been
made by Sierra (2000) and Moerdijk (2008). There are also various collocation
dictionaries (BBI, OECD), reverse dictionaries (Bernstein, 1975; Kahn,
1989; Edmonds, 1999) and *OneLook*, which combines a dictionary (*Word-
Net*) and an encyclopedia (*Wikipedia*). 5 Finally, there is *MEDAL* (Rundell
and Fox, 2002), a thesaurus produced with the help of Kilgariff’s *Sketch
Engine* (Kilgariff *et al*., 2004). A lot of progress has been made over the
last few years, yet more can be done especially with respect to indexing (the
organization of the data) and navigation. Given the possibilities modern
computers offer with respect to storage and access, computational lexicog-
raphy should probably jettison the distinctions between lexicon, encyclope-
dia, and thesaurus and unify them into a single resource.

One detail often overlooked is the fact that in searching the dictionary
speakers rarely access the desired information directly. With arguably the
sole exception of spelling-checking (e.g., neighbor vs. neighbour), speakers
generally proceed via some intermediate term, resorting to lexical entries
(headwords) when seeking meaning 6 or some grammatical information
concerning a given word. 7 Speakers provide meanings (possibly elements of
the definition or description of a word: ‘arboreal,’ ‘herbivorous,’ ‘marsupial,’
‘native of Australia’) or semantically related words (synonyms, antonyms,
hyponyms, etc.; see also Figure 5 in Section 2) when looking for the corre-
sponding lexical form (in this case, ‘koala’). In sum, dictionaries serve many

---


6 Imagine not knowing the meaning of ‘mitre’ or the differences between closely
related words (‘caiman,’ ‘crocodile,’ ‘alligator’).

7 Suppose one wanted to know whether the French say ‘le lune’ or ‘la lune,’ when
referring to the moon (lune).
functions. Since our concern is language production, we are interested in the problem of how authors (speakers/writers) can specify their needs to get the linguistic form (usually a headword or lemma) for an idea or a concept they want to, but cannot, express due to an inability to access the corresponding word. This point will be addressed below in Section 2.

In this article, we pursue the following two goals: (a) to show that even computers may fail to access stored information; (b) to describe a way to help people access a momentarily unavailable word. To this end we will explain some design features of a dictionary suitable for the language producer.

1.2 Storage does not guarantee access

To test this claim we ran a small experiment, comparing an extended version of WordNet (WN), eXtended WN (Mihalcea and Moldovan, 2001) and Wikipedia, which we converted into a lexical resource. Our goal was not so much to check the quality of WN or any of its extensions\(^8\) as to show, firstly, that storage does not guarantee access and, secondly, that access depends on the quality of the resource within which the search takes place, the index, and the type of query. Having two resources built with different foci, our goal was to check the efficiency of each with respect to word access. For practical reasons (limitation of processing time) we considered only direct neighbors (i.e. words at a distance of 1). Hence, we defined a function called direct neighborhood (henceforth $f_{dngh}$), which, once applied to a given window (sentence/paragraph),\(^9\) produces all its co-occurences. Of course, what holds for direct associations (our case here), holds also for indirectly related words (distance >1), that is, mediated associations.

1.2.1 Using WordNet as a corpus

WN is a lexical database for English, developed under the guidance of G.\(^8\) We used WN 2 rather than the most recent version WN 3. Nevertheless, the claim we make here applies to all dictionaries, regardless of their size. The weak point does not lie in the data (lexical coverage), but in the (lack of) quality of the index (structure of the dictionary).

\(^9\) Optimal size is an empirical question, which may vary with the text type (encyclopedia vs. raw text).
Miller, a well-known cognitive psychologist (Miller et al., 1990). One of his goals was to build a resource resembling the mental lexicon (associative network), allowing navigation akin to the human mind by travelling along the pathways of the network (spreading activation). The structure of his resource is quite different from conventional, alphabetically organized dictionaries. Also, rather than multiplying the number of dictionaries, —one for each use or task (to find a definition, synonym or antonyms, etc.),— WN was built as a single resource, allowing access via multiple paths and different links. Since WN is very well known, we will not describe it here.10

While WN is a lexical resource, it can also be seen as a corpus. This can be very useful, especially if one wants to compare it with other corpora like Wikipedia (Wp)11 or if one wants to make use of a specific part of the base, for example, glosses. Since glosses correspond roughly to a word’s meaning (definition), their elements (bag of words) can be used for accessing the word form (headword).

WN has had a strong impact in the NLP community12 where it is heavily used. This has led to the creation of various extensions. As mentioned in the foregoing, we use one of them, eXtended WN (Mihalcea and Moldovan, 2001, Harabagiu and Moldovan, 1998) since it spares us the trouble of having to deal with problems inherent to automatic text processing: segmentation (establishment of word boundaries), ambiguity resolution, lemmatization (deflection of words), etc.

Despite all this, two important problems remain: the size of the corpus (our version contains only about 144.000 words) and the lack of encyclopedic knowledge, i.e. syntagmatic associations, which, taken together may impede

10 The reader not familiar with this groundbreaking work may consult the WordNet website (http://wordnet.princeton.edu/).
11 Wikipedia is a collaboratively built, multilingual encyclopedia, freely accessible on the internet (http://www.wikipedia.org). The collaborative building of such resources (dictionaries, encyclopedias, etc.) has become quite popular over the last few years. For example, there have been two workshops devoted the collaborative building of semantic resources and their role and influence in NLP (ACL, 2009; COLING, 2010).
12 Strangely, it did not have the same success in circles dealing with language production, neither among psychologists, nor in the NLG community.
Memorisation of Words does not Guarantee Their Access

239

WordNet
does not Guarantee Their Access

239

Figure 1. WordNet as corpus (example: ‘ant’).

word access (see below). Indeed, concepts that are functionally related in normal life (‘to dine-table-meal’; ‘to fish-net-tuna,’ etc.) should evoke each other, yet this is not really the case in $WN$.\footnote{This problem has been recognized by the authors as the tennis problem (Fellbaum, 1998), meaning that topically related words are not stored together or cross-referenced. Hence, ‘ball,’ ‘racket,’ and ‘umpire’ appear on different branches of the hierarchy, even though all of them could be part of the very same topic, “the game of tennis.” Likewise, ‘instrument’ and ‘used_for,’ appear in different parts of the resource, though they are (quasi-) synonyms. It should be noted though that efforts have been made to overcome these problems. For example, information may be found in the glosses (in the case of ‘used_for’ and ‘instrument’), and topically related words may be accessed now to some extent (Boyd-Graber et al., 2006).}

1.2.2 Using WikiPedia as a corpus

In order to compare $WP$ with $WN$, we used the English version, which at the moment of writing this paper (September, 2011) contained 3,550,567 entries. It is on the basis of this set that we built our resource. $WP$ has exactly the opposite properties of $WN$. While it contains many encyclopedic relations, i.e. syntagmatic associations, it is only raw text. Hence, problems like text segmentation, lemmatisation and stoplist need be addressed. To avoid these problems we used Dbpedia (Bizer et al., 2009), a plain text version of $WP$, where all HTML tags have been removed. Running a part-of-speech tagger\footnote{In this experiment, we used our own part-of-speech tagger based on the English dictionary of forms, the DELA (http://infolingu.univ-mlv.fr/DonneesLinguistiques/} allowed us to annotate the major elements of the
paragraph and to filter out all irrelevant words, to keep but the most important ones (nouns, adjectives, verbs and adverbs). These latter were then used for building our database.

1.3 Exploitation and comparison of the resources

Constructing the resource requires the processing of a corpus and the building of a database. To this end we used a corpus applying our direct neighborhood function $f_{\text{ing}}$ to a predetermined window (a paragraph in the case of encyclopedias). The result (i.e. the co-occurrences) are stored in the database, together with their weight, (i.e. number of times two terms appear together) and the type of link. As mentioned above, this kind of information is needed later on for ranking and navigation.\textsuperscript{15}

At present, co-occurrences are stored as triplets $(S_w, T_w, \text{times})$, where $S_w$ and $T_w$ refer respectively to the query word (i.e. trigger- or source word) and target word, term obtained in response to the query (direct association), while times represents the weight (i.e. the number of times the two terms co-occur in the corpus), the scope of co-occurrence being the paragraph. Of course, there are many other ways to determine the weight, but this shall not concern us here.

1.3.1 Usage

To show the relative efficiency of a query (see Figure 3), we have developed

\[ \text{Dictionnaires/telechargement.html) } \]

\textsuperscript{15} This latter aspect is not implemented yet, but will be added in the future, as it is a necessary component for easy navigation (see Section 2.2.3).
Welcome to the WORDFINDER webpage

**Input**

```
harvest wine grapes
send
```

**Output** (found, related words): 23 hits

```
Beaujolais, regions, area, quality, between, vintage, well, usually, vineyards, south, various, year, growing, early, cru, low, north, following, aging, generally, time, potential, very
```

Figure 3. Outputs produced in response to the inputs « harvest, wine, grapes »

a website in Java as a servlet which will soon be released on our respective homepages. Usage is quite straightforward: people add or delete a word from the current list, and the system produces some output. For example, if the inputs were “harvest, wine, grapes,” the system would display all co-occurring words (i.e. direct associations). Of course, if we use more than one corpus, as we do, we will have to display the results for each one of them.

The output is an ordered list of words, the order of which depends on the overall score (i.e. the number of co-occurrences between the $S_w$ and the directly associated word, called the ‘potential target word’ (PT$_w$)). For example, if the $S_w$ ‘bunch’ co-occurred five times with ‘wine’ and eight times with ‘harvest,’ we would get an overall score or weight of 13: ((wine, harvest), bunch, 13). Weights can be used for ranking (i.e. prioritizing words) and the selection of words to be presented, both of which may be desirable when the list becomes long.

1.3.2 Examples and comparison of the two corpora

Listed below in Figure 4a are the results produced by *eXtended WN* and *WP* for the following inputs: ‘wine,’ ‘harvest’ or their combination (‘wine + harvest’).
Our goal was to find the word ‘vintage.’ As the results show, ‘harvest’ is a better query term than ‘wine’ (488 vs 30 hits), and their combination is better than either of them (6 hits). What is more interesting is the fact that none of these terms allows us to access the target even though it is contained in the database of $\textit{WN}$, which clearly supports our claim that storage does not guarantee access (see also, Sinopalnikova and Smrz, 2006; Tulving and Pearlstone, 1966).

Things are quite different if we build our index on the basis of information contained in $\textit{WP}$. The same input, ‘wine’ evokes many more words (3045 as opposed to 488, with ‘vintage’ in the 81st position). For ‘harvest’ we get 4583 hits instead of 30, ‘vintage’ occurring in position 112. Combining the two yields 353 hits, which pushes the target word to the third position, which is close to the top of the list.

We hope that this example is clear enough to convince the reader that it makes sense to use real text (ideally, a well-balanced corpus) to extract from it the information needed (associations) in order to build indexes allowing users to find the words they are looking for.

### 1.3.3 Analysis of this relative failure

One may wonder why we failed to access information contained in $\textit{WN}$ and why $\textit{WP}$ performed so much better. We believe that the relative failure of

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>wine</td>
<td>488 words: grape, sweet, serve, france, small, fruit, dry, bottle, produce, red, bread, hold...</td>
<td>3045 words name, lord characteristics, christian, grape, France, ... vintage (81st), ...</td>
</tr>
<tr>
<td>harvest</td>
<td>30 words month, fish, grape, revolutionary, calendar, festival, butterfish, dollar, person, make, wine, first,...</td>
<td>4583 words agriculture, spirituality, liberate, production, producing, ..., vintage (112th), ...</td>
</tr>
<tr>
<td>wine + harvest</td>
<td>6 words make, grape, fish, someone, commemorate, person, ...</td>
<td>353 words grape, France, vintage (3rd), ...</td>
</tr>
</tbody>
</table>

**Figure 4a.** Comparing two corpora with various inputs.
Memorisation of Words does not Guarantee Their Access

$WN$ is mainly due to these two facts: the size of the corpus (114,000 words as opposed to 3,550,000 for $WP$), and the number of syntagmatic links, both of which are fairly small compared to $WP$. Obviously, being an encyclopedia, $WP$ contains many more syntagmatic links than $WN$. Of course, one could object that we did not use the latest release of $WN$ (version 3.0) which contains many more words (147,278 words, clustered into 117,659 synsets), but still, this would not fundamentally affect our line of reasoning or our conclusion. Even in a larger set we may fail to find what we are looking for because of the resource, the quality of the query, or both. Moreover, as mentioned already, the weak point is not so much the quantity of the data, as the quality of the index (the relative sparsity of links).

Nevertheless, in order to be fair towards $WN$, one must admit that, had we built our resource differently, for example, by including in the list of related terms, not only the directly evoked words (i.e. potential target words), but all the words containing the source-word (wine) in their definition (Bordeaux, Retsina, Tokay), then we would certainly get ‘vintage,’ as the term ‘wine’ is contained in its definition (‘vintage’: a season’s yield of ‘wine’ from a vineyard). Another noteworthy point is the fact that success may vary quite dramatically, depending on the goal. As Table 4b shows, $WN$ outperforms $WP$ for the words ‘ball,’ ‘racket’ and ‘tennis.’ Yet, $WP$ does not lag much behind; additionally, it contains many other words possibly leading to the target words (“player, racket, court,” ranked, respectively as numbers 12, 18 and 20). Not being an encyclopedia, $WN$ lacks most of them, though

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ball</td>
<td>346 words</td>
<td>4891 words</td>
</tr>
<tr>
<td></td>
<td>game, racket, player, court, volley, wimbledon, championships, inflammation, ..., tennis (15th), ...</td>
<td>sport, league, football, hand, food, foot, win, run, game, ..., tennis (27th), ...</td>
</tr>
<tr>
<td>racket</td>
<td>114 words</td>
<td>2543 words</td>
</tr>
<tr>
<td></td>
<td>break, headquarter, gangster, lieutenant, rival, kill, die, ambush, tennis (38th), ...</td>
<td>death, kill, illegal, business, corrupt, ..., tennis (72nd), ...</td>
</tr>
<tr>
<td>ball + racket</td>
<td>11 words</td>
<td>528 words</td>
</tr>
<tr>
<td></td>
<td>game, tennis, (2nd), ...</td>
<td>sport, strike, tennis (3rd), ...</td>
</tr>
</tbody>
</table>

**Figure 4b.** Comparing two corpora with various inputs.
surprisingly, it contains ‘Seles’ and ‘Graf,’ the names of two great female tennis players. However, given the respective qualities of WN and WP one may well consider integrating the two by relying on a resource like WordNet++.%

In the remainder of this paper we will briefly sketch our roadmap for building a dictionary for the language producer.

2. A dictionary for the language producer

Obviously, dictionaries for the writer ought to be different from dictionaries for the reader with regard to the input, the structure, the index, and search facilities. While a reader may access the word’s meaning directly via its corresponding linguistic form,— the headword ‘dog’ yielding something like ‘pet’ or ‘member of the genus Canis,’— the opposite does not hold true. Neither the inputs ‘pet,’ nor ‘member of genus canis’ would give us ‘dog’ and only this word, especially if the dictionary is large. No input will ever be precise enough to yield but the desired word. Second, even if the input did describe perfectly well the target word, the user would still be given more than one word (synonym set), hence he would have to choose, say between, big, large, huge, etc. Third, in most cases we do not even know in what terms to characterize the target word. Hence we proceed by approximations, which implies that we access the target word indirectly, i.e. via some intermediate term(s).

There are at least three things that authors know when looking for a specific word: its meaning (definition) or at least part of it (this is the most frequent situation), its lexical relations (hyponymy, synonymy, antonymy, etc.), and the collocational or encyclopedic relations it entertains with other words (Paris-city Paris-French capital, etc.). Put differently, there are several ways to access a word (see Figure 5): via its meaning (concepts, meaning fragments), thesaurus- or encyclopedic relations, form relations (spelling, sound), lexical relations, syntactic patterns (search in a corpus), and, of course, via translations. While we will mainly draw on encyclopedic relations (mostly syntagmatic associations), we have presented elsewhere (Zock

---

% WordNet++ is available at http://lcl.uniroma1.it/babelnet/.
Memorisation of Words does not Guarantee Their Access

et al., 2010) the building principles of a lexical matrix which integrates all of these possibilities (except the last one and the clang relation) into a single resource.

However, people seem to know more than that. Psychologists who have studied people in the tip-of-the-tongue state (Brown & McNeill, 1966; Vigliocco et al., 1997) have found that their subjects had access not only to meanings (the word’s definition), but also to information concerning gram-

17 This feature of the mental lexicon (ML) is very important since if one method fails, it allows us to resort to another. It is also worth noting that a thesaurus and an encyclopedia can have an impact at different levels, the level of ideas (brainstorming, conceptual level) and the level of words (linguistic level). Hence, a thesaurus can be used for conceptualization, that is, message specification (meaning) as well as for its expression (lexicalization). Indeed, authors often start from a broad concept (ANIMAL), which they gradually narrow down (reptile), before committing to a specific lexical form: alligator/crocodile/caiman. Likewise, encyclopedic relations (associations) may be used for message creation, as well as for finding a concrete lexical form. Concepts evoking (other) concepts (red-fire) and words priming other words (coffee-strong). In both cases the same mechanism is at work, though at different levels and operating on different elements: concepts in one case, words in the other.
mar (gender) and lexical form: the number of syllables, the beginning/ending of the eluding word, the part of speech. While all this information could be used to constrain the search space, —the ideal dictionary being multiply indexed,— we will deal here only with the target word’s meaning (definition, bag of words), lexical relations and otherwise semantically related words (associations, collocations in the large sense of the word). Before discussing how such a dictionary could be built and used, let us consider a possible search scenario.

2.1 A possible search scenario
When looking for a word, people tend to start from a close neighbour. For the sake of the argument, let us assume that they cannot think of a directly connected word, the only token coming to their mind being a word they know to be somehow connected to the $T_w$. Suppose, one were to express the following ideas: *superior dark coffee made from beans from Arabia*, knowing that neither ‘espresso’ nor ‘cappuccino’ are the intended word. While none of this would lead you directly to the desired word, ‘mocha,’ the information at hand (i.e. the word’s definition or some of its elements) could certainly be used. In addition, people draw on knowledge concerning the *role* a concept (or word) plays in language and in the real world (i.e. the associated terms it evokes). For example, they may know that they are looking for a noun standing for a *beverage* that *people* take under certain circumstances, that the *liquid* has certain properties, etc. In sum, people have in their mind an encyclopedia with all these words, concepts, or ideas being highly connected. Hence, any one of them has the potential to evoke the others. The likelihood for this to happen depends, of course, on factors such as frequency (associative strength), recency (last occurrence), distance (direct vs. indirect access), prominence (saliency), etc.

Let us see how this could work. Suppose one was looking for the word ‘mocha’ (target word: $T_w$), yet the only token coming to mind was ‘Java’ (query- or source word: $S_w$).\(^\text{18}\) Taking the latter word as the starting point,

\(^{18}\) Note that this, just like many other homonyms, might lead us to a completely different domain: *island* vs. *programming language*. In other words, homonyms can be considered as a shortcut in the network.
Memorisation of Words does not Guarantee Their Access

247

the system would show all the connected words. However, being ambiguous, —‘Java’ is a homonym, referring either to an island or to a *programming language*,— the user has to specify which of the two she has in mind. Since the programming language seems to have no connection to the final goal,¹⁹ she will choose Java, the island. This latter has various connections (not shown in Figure 6) concerning geography, language, religion and local products (coffee, rice, etc.). Since this last link is compatible with the user’s goal,— (who, even though she could not name it, knows subconsciously that she is looking for some kind of drink),— she will choose ‘coffee’ and finally ‘mocha,’ a type of ‘beverage’ made from such beans.

Of course, the word ‘Java’ might just as well trigger ‘Kawa’ which not only rhymes with the $S_w$, but also evokes ‘Couawa,’ also written ‘Kawa,’ an argotic word of coffee in French, or ‘Kawa Igen,’ a javanese volcano. Last, but not least, the user could well have started with any other coffee producing country, unless she would start right away from a closely related associate of mokka, namely, a type of coffee or beverage made out of beans (probably the most frequent starting point). However, we wanted to show

¹⁹ Note that ‘Java beans,’ a notion inherent to *JAVA*, the programming language, could get the user back on track, leading her to the desired target word expressing a specific beverage made out of coffee beans (mocha).
precisely that even if one does not start from a direct neighbour, it generally
takes only a few mouse clicks to get from the $S_w$ to the $T_w$.

Since everything is connected, words can be accessed via multiple routes.
Also, while the distance covered in our example is quite unusual, it is pos-
sible to reach the goal quickly. It took us actually very few moves, to find
a connection between the word ‘Java’ and ‘coffee beans’ or the drink made
out of them. Of course, ‘cybercoffee’ fans, people thinking of a particular
port city on the Red Sea coast of Yemen (Mocha), or people familiar with
the fact that Melville’s novel *Moby Dick* was inspired by Mocha Dick, a
white whale often found at Mocha, a Chilean island in the Pacific, might be
even quicker in reaching their goal.

2.2 Building the semantic map
Search usually takes place somewhere, in some kind of space, and in order
to help people find their way, we have built a map. Such maps can be built
for many things, including a lexicon. Since our maps are mainly based on
semantic information, we refer to them as semantic maps. Lexical graphs
are such maps, with words being nodes, and the links being the roads lead-
ing from one word to the other. This is the fundamental structure of the net-
work. In addition, links are qualified or typed (isa, synonym, etc.), and they
are quantified (i.e. weighted).

Of course, there are various methods to build such a map. One way is to
obtain lists of associations by asking people (Deese, 1965). This has been the
main strategy of the psychologists who have built word association norms
(Nelson *et al.*, 1998). Another way is to use games (Lafourcade, 2007). Still
another approach is corpus-based, by extracting collocations. This is our
strategy. There are several problems that need to be addressed.

- **Building a representative corpus.** We need a well-balanced corpus.
  Since the corpus is supposed to represent the user’s world knowledge,
  this knowledge must be reflected in the corpus. In other words, the cor-
  pus must contain a little bit of everything normal people know concern-
  ing the world in general (objects and relations), but also specific inform-
  ation concerning recent events (sports, politics, etc.).
- **Indexing.** Words have to be indexed. We do this in terms of the asso-
Memorisation of Words does not Guarantee Their Access

To this end we need to discover the trigger words and their collocates. Psychologists built such lists decades ago (Deese, 1965; Schvaneveldt et al., 1989). Similar lists are now freely available on the web. For example, there is the Edinburgh Associative Thesaurus\textsuperscript{20} and the compilation of Nelson \textit{et al.}\textsuperscript{21} There are also resources for various languages, for example, German,\textsuperscript{22,23} and Japanese.\textsuperscript{24} In addition to using these data directly, one may try to build this kind of resource bottom-up, via corpora and by using a collocation extractor.

- **Ranking.** The weight of the linked words needs to be determined (relative frequency). This is important for ranking the associated words. Ideally, the weight is (re-)computed on the fly to take into account contextual variations. A given word, say ‘Java,’ is likely to evoke quite different associations depending on the context: ‘tourism’ vs. ‘programming.’

- **Identification** and ‘typing’ of the links. Associations must not only be identified, they must also be labeled. Qualifying (i.e. typing the links) is the hardest task, yet it is vital for navigation. Frequency alone is not only of limited use, —(people cannot interpret properly numerical values in a context like this), — it is even misleading: two terms of very similar weight (say, ‘mouse’ and ‘PC’) may belong to entirely different categories (‘rodent,’ ‘computer device,’ ‘type of computer’). Hence, choosing one instead of the other may have important consequences concerning the chances of finding the desired target word.

2.3 Navigating in the resource by using a lexical compass

Once this resource is built, access is straightforward. The user gives as input a word he believes to be directly or indirectly connected to the Tw,\textsuperscript{25} say ‘hospital,’ to which the system would answer with all immediate associates (‘clinic,’ ‘sanatorium,’ ‘doctor’). If the list contains the Tw, the search stops,

\textsuperscript{20} http://www.eat.rl.ac.uk/
\textsuperscript{21} http://cyber.acomp.usf.edu/FreeAssociation
\textsuperscript{22} http://www.schulteimwalde.de/resource.html
\textsuperscript{23} http://www.coli.uni-saarland.de/projects/nag/
\textsuperscript{24} http://www.valdes.titech.ac.jp/~terry/jwad.html
\textsuperscript{25} This is, of course a simple case. One could also think of several terms as input.
otherwise it continues. The user chooses a word from the list (say, ‘doctor’) or a word being evoked by them (indirect association), and the system will reveal again all directly associated terms: ‘surgeon,’ ‘pediatrician,’ ‘medic.’

This problem cannot be solved via the well-known algorithm computing the shortest path between two nodes in a network, as this supposes that one knows both points. In our case we know only the starting point (user query), but not the end point (goal); if we knew it, there would be no need for search to begin with, we would just display the $T_w$. Nevertheless, even though the user does not know the $T_w$, he can recognize it (goal) if he sees it in a list.\footnote{This kind of passive knowledge is somehow akin to the tip-of-the-tongue state. As psychologists (Brown and McNeill, 1966) have shown, people in this state have a lot of information concerning the $T_w$, to the point that, if this latter is presented to them, they can recognize it without ever making any mistakes.} He can do even more. If, following his query ($S_w$), we give him a set of words, he will (arguably) know in most cases which one of them is the most promising one (i.e. the one closest to the $T_w$). As the reader can see, search is interactive. The user provides the input (query, or $S_w$), the direction to go if none of the presented words corresponds to the $T_w$, and when to stop (successful search). The system provides hints (words, potentially leading to the $T_w$), in the event that none of the words corresponds to the $T_w$ (mediated, i.e. indirect associations between the $S_w$ and the $T_w$).

Obviously, there are some important differences between a conventional compass and our navigational tool. While the former automatically points to the north, letting the user compute the path between his current location and the desired goal (destination), the latter assumes the user knows the goal or at least its direction. While the user cannot name the goal (she has only passive knowledge), the system cannot guess it. However it can make valuable suggestions. In other words, the system can give hints concerning potential goals, but it is nevertheless the user who decides on the direction to go, as only she knows which suggestion corresponds to the goal, or which one of them is the most closely connected.

\textbf{2.4 Potential interface problems}

Since words occur in many different settings or syntactic contexts, every
Memorisation of Words does not Guarantee Their Access

Figure 7. Potential problems with graphs: crossing links with indirect neighbours.27

...word is likely to have a rich set of connections. Obviously, the greater the number of words associated with a term, and the more numerous the type of links, the more complex the graph will be. This reduces considerably their interface value and their potential to support navigation. There are at least three factors impeding readability:

• **High connectivity** (i.e. the great number of possible links). These links can be of different types, bi-directional, asymmetric and of different weights.

27 IS-A (subtype); SYN (synonym); TIORA (‘Typically Involved Object, Relation or Actor,’ for example, tools, employees).
• **The possible crossing of links** in the case of indirect association (see \(A_1 - B_2\) or \(A_2 - B_1\) in Figure 7).\(^{28}\)

• **Distribution** (i.e. non-adjacency, of conceptually related nodes, that is, nodes activated by the same kind of association (e.g. synonyms), but not being displayed next to each other (see Figure 7, \(A_1 - A_3\) or \(A_2 - A_4\)). This is confusing to the user.

### 2.5 A possible solution

We believe that there is a fairly straightforward solution to the problem. Since all words are connected via one-directional links, we have a graph in which everything can be reached from anywhere,\(^{29}\) regardless of the starting point. However, our graph can also be seen as a set of trees. Since a search could be launched at any point, —any node of the graph could become the query or starting point (i.e. the root of the tree),— we have as many trees as the graph contains nodes. Hence, the input (Sw, say ‘hospital’) would become the root of the tree, and the associated terms (i.e. output or potential target words) would be the leaves: ‘clinic,’ ‘sanatorium,’ ‘doctor,’ ‘nurse.’ If the input and output are linked via different kinds of association (‘hospital-clinic’ vs. ‘hospital-ambulance’), we create an intermediate node for each link, giving it the name of the link (‘subtype’ vs. ‘part-of’). Put differently, we create as many nodes as there are different kinds of links emanating from a given node.

In conclusion, rather than displaying all the connected words as a graph or as a huge flat list, we display them in hierarchically organized *categorial*

---

\(^{28}\) Note, that the crossing of lines can be avoided in the immediate neighborhood (distance 1, i.e direct associations), but not at the next level. If two sets of words, say A1 and A3, on the one hand, and A2 and A4, on the other, have at the next level B1 and B2 as associates, then the links are bound to cross. Also, bear in mind that the scope is the entire graph and not only the next adjacent level (i.e. direct neighbors). Note also, that this crossing of links is a side-effect of mapping an n-dimensional graph on two dimensions.

\(^{29}\) While our semantic memory contains bi-directional links —(if A evokes B chances are, that B also evokes A.),— we assume that during search activation spreads only in one direction. If our starting point is A, it will evoke B, since it is a directly connected word.
Memorisation of Words does not Guarantee Their Access

clusters, factorizing links, so that the nodes now present all words having the same kind of relationship with respect to the Sw. Put differently, we suggest to display by type or category (chunks) all words entertaining a specific link with respect to the input or the Sw (see Figure 8). Since the same word may occur at various levels, our trees allow for recovery (see our example of ‘Java beans’ mentioned in footnote 19). If one has taken a wrong turn at some point, one may still reach one’s goal via a detour. Of course, we will

30 For example, a word like ‘coffee’ may be connected both to ‘beverage’ and to ‘export product.’ Since words may participate in various scenarios (‘cat-chase-mouse,’ ‘cat-play_with-mouse’ vs. ‘cat-cute’), they are connected via different kind of associations. Hence, a lemma may be accessed via different paths (i.e. it will be revealed at different points in the tree).

---

**Figure 8.** Search-tree with two links: IS-A (subtype of) and TIORA (Typically Involved Object, Relation or Actor).
not display the entire tree, but only level by level the set of associated leaves and nodes for a given query (i.e. the choice of a word or choice of a node).

This kind of presentation seems clearer and less overwhelming for the user than graphs, as it allows for categorical search, which is a lot faster than searching in a huge bag of words. Of course, this supposes that the user knows to which category a word belongs to, and that the labels (i.e. link names) are well chosen. This is crucial, as the names must be meaningful (i.e. interpretable by the user), which may be problematic in our example here. Figure 8 presents visually the rationale outlined above (Section 2.1), except that the Sw and Tw are not ‘Java’ and ‘mocha’ but ‘hospital’ and ‘medic.’

As one can see, the fact that the links are labeled has some very important consequences. First, while maintaining the power of a highly connected graph (possible cyclic navigation), it has at the interface level the simplicity of a tree: each node points only to data of the same type, (i.e. to the same kind of association). Second, with words being presented in clusters, navigation can be accomplished by clicking on the appropriate category. The assumption being that the user either knows to which category the target word belongs or can recognize within which of the listed categories it falls. It is also assumed that categorical search is in principle faster than search in a huge list of unordered (or, alphabetically ordered) words.

3. Conclusion and Future Works

We began this article by challenging the widely held assumption that stored information can always be accessed. Comparing two resources different in size and quality led to the conclusion that one needs a great deal of data of a various kind (encyclopedic, news) in order to index properly the lexical resource. While indexes are vital for accessing data, there are many ways of building them. Since the ultimate users will be humans, the index should reflect their habits to classify objects or perceived relations between them.

Obviously, for a dictionary to be truly useful it must not only contain an abundance of information but must also reveal it when needed. To this end we suggest to index the data according to various points of view. This will yield a network in which anything is reachable from anywhere regardless of the starting point. As we know from every day life, and as psychologists
Memorisation of Words does not Guarantee Their Access

studying the tip-of-the-tongue problem have shown (Brown and Mc Neill, 1996), whenever we search for something we have stored, say, a word, we can access some kind of information concerning the target object. This kind of information should be utilized. However, knowledge is variable, changing from person to person, and moment to moment. Hence our tools should be able to accommodate this fact.

In order to capture the kind of information needed and in order to build the resource just described, we propose to take a mix of corpora and to extract both stable as well as dynamic knowledge. While the former is more likely to be found in encyclopedias and books containing more or less universally shared information, the latter is more local and often found in the news.

Among the many issues to be addressed is the building of a prototype for a small domain to see whether or not our intuitions hold, the final judge being of course the user. Among the problems that we have to address are the notion of links (there is no satisfying list available at the moment) and the problem of interpreting user queries. Obviously, a given query will have different meanings depending on the moment of its usage. For example, in the ‘mocha’ example, the user giving ‘island’ as key expects specific information concerning the ‘island Java’ and not just any island, or ‘islands’ in general.

While more work is needed, we do believe that it is worth the effort, for through this endeavor one will learn something about the functioning of the human mind and the structure of the mental lexicon. In addition, the problems addressed (indexing and navigation) go well beyond language and deal as well with human memory, i.e. how information is structured, indexed and stored.

References


Goldman N. 1975. Conceptual Generation, in Schank, R. (ed.): Conceptual Informa-
Memorisation of Words does not Guarantee Their Access


