Eureka! A Simple Solution to the Complex 'Tip-of-the-Tongue'-Problem

Michael Zock

Chapter 14
Eureka! A Simple Solution the Complex ‘Tip-of-the-Tongue’-Problem

Michael Zock

To search for a word in a dictionary without a proper index *is like looking for an address in a* city without a decent map.

Abstract Dictionaries are repositories of knowledge concerning words. While readers are mostly concerned with *meanings*, writers are generally more concerned with *word forms* expressing meanings (lemma). I will focus here on this latter task: building a tool to help authors to find the word they are looking for, word they may know but whose form is eluding them. Put differently, my goal is to build a resource helping authors to overcome the **T**ip-of-the-**T**ongue problem (ToT). Obviously, in order to access a word, it must be stored somewhere (brain, resource). Yet this is far from sufficient. Access may depend on many other factors than storage of word forms: organization of the dictionary (index), the user’s cognitive state, i.e. available knowledge at the onset of search, the distance between the source- and the target-word (direct neighbor or not) , the knowledge of the relationship between the two, etc. I will try to provide evidence for the claim that (word) storage does not guarantee access. To this end I will compare a well-known lexical resource, WordNet (WN), to an equivalent one, but bootstrapped from Wikipedia (WiPe). While both versions contain basically the same set of words, the latter contains many more (syntagmatic) links than WN. This is probably the reason why WiPe outperforms WN. In the last two sections I will explain under what conditions WN is suitable for word access, and what it might take to go beyond the limitations of this famous resource.

M. Zock (✉)
Aix-Marseille Université, CNRS, LIF (UMR 7279), 163, Avenue de Luminy,
13288 Marseille, France
e-mail: michael.zock@lis-lab.fr

À. Massip-Bonet et al. (eds.), *Complexity Applications in Language and Communication Sciences*, https://doi.org/10.1007/978-3-030-04598-2_14

14.1 Introduction

Speaking a language can be a daunting task. Planning *what to say* (message) and *how to say it* (linguistic form) have to be carried out on the fly, that is, quasy simultaneously, and while finally *saying* it (articulation) one may have to plan already the next stretch of discourse. Actually, speaking is quite a bit more complex than that, requiring the solution of at least half a dozen of problems: determine content, find suitable words and sentence frames, add function words, perform morphological operations (agreement) and articulate.

To get a better idea of the complexity of the task at hand let's focus only on one of them, lexical access. Bear in mind though that speech is fast (2–5 words per second, Levelt 1993, 2001), and that words must be found in a resource containing, say, 100,000 words (Miller 1991). This raises a number of interesting questions:

- how do people manage to access words so quickly in such a huge lexicon, succeeding most of the time while making very few mistakes?
- can the insights gained by studying the *mental lexicon* (storage, organization, process) be transposed to an external resource? Put differently, can we use this knowledge to enhance paper- or electronic dictionaries (off-line processing)?
- If ever the answer is negative. Are there ways to achieve something equivalent, at least in terms of precision, that is, can we build or enhance an existing resource (electronic dictionary) in such a way as to allow its users to find quickly the desired word?

Having addressed the first two questions elsewhere (Zock et al. 2010), I will focus here mainly on the last problem, building a resource meant to help users to overcome the *tip-of-the-tongue problem*.[1] Hence, functionally speaking I try to achieve something equivalent to the human brain, though in slow motion: help people to find the word they are looking for. Before showing the way how this can be done, let me say a few words concerning the word-access problem.

When speaking or writing we encounter basically either of the following two situations: one where everything works automatically (Segalowitz 2000), somehow like magic, words popping up one after another as in a fountain spring, leading to a discourse where everything flows like in a quiet river (Levelt et al. 1999; Rapp and Goldrick 2006). The other situation is much less peaceful: discourse being hampered by hesitations, the author being blocked somewhere along the road, forcing her to look deliberately and often painstakingly for a specific, possibly known word (Zock et al. 2010; Abrams et al. 2007; Schwartz 2002; Brown 1991). I will be concerned here

[1] The ToT problem is characterized by the fact that the author has only partial access to the word form s/he is looking for. The typically lacking parts are phonological (Aitchison 2003). The ToT problem is a bit like an incompleted puzzle, containing everything apart from some minor small parts (typically, syllables, phonemes). Alas, not knowing what the complete picture (target, puzzle) looks like, we cannot determine the lacking part(s). Indeed, we cannot assume to know the target, and claim at the same time to look for it or any of its elements. Actually, if we knew the target (word) there wouldn't be a search problem to begin with, we would simply spell out the form.

with this latter situation. More specifically, I am concerned here with authors using an electronic dictionary to look for a word. While there are many kind of dictionaries, most of them are not very useful for the language producer. The great majority of them are *semasiological*, that is, words are organized alphabetically. Alas, this kind of organisation does not fit well the language producer whose *starting points* (input) are generally meanings[2] or cue-words (primes) and only the *end point* (outputs) the corresponding target word.

To be fair though, one must admit that great efforts have been made to improve the situation both with respect to lexical resources and electronic dictionaries. Since the invention of the thesaurus (Roget 1852) quite a few *onomasiological dictionaries* have been built (van Sterkenburg 2003; Casares 1942), even if nowadays they are not built by hand anymore (Dornseiff et al. 2004; Rundell and Fox 2002). Today we build lexical resources via corpora (Hanks 2012; Kilgarriff and Kossem 2012), crowd-sourcing (Benjamin 2014), dictionary writing systems (Abel 2012) and corpus tools like Sketch Engine (Kilgarriff et al. 2004), MonoConc or WordSmith. Yet thesauri are not the only kind of onomasiological resources. There are *analogical dictionaries* (Boissière 1862; Robert et al. 1993), *collocation dictionaries* (Benson et al. 2010), *reverse dictionaries* (Bernstein 1975; Kahn 1989; Edmonds 1999), *rhyme dictionaries* (Fergusson and Fergusson 1985 ; Webster 2007), and *network-based* lexical resources: *WordNet* (Fellbaum 1998; Miller 1990), *MindNet* (Richardson et al. 1998), *HowNet* (Dong and Dong 2006), and *Pathfinder* (Schvaneveldt 1989). There are Longman’s *Language Activato*r (Summers 1993) and *OneLook*[3], which, akin to BabelNet (Navigli and Ponzetto 2012), combines a dictionary (WordNet) and an encyclopedia (Wikipedia). Besides all this there are interesting proposals coming from Fontenelle (1997), Sierra (2000), Moerdijk (2008), and Mel’çuk (Mel’çuk and Polguère 2007). Finally, there is MEDAL (Rundell and Fox 2002), a thesaurus produced with the help of Sketch Engine (Kilgarriff et al. 2004).

In parallel to dictionary making a lot of progress has been made by psycholinguists who study the time course of lexical access (Levelt et al. 1999), word associations (de Deyne and Storms 2015) and the structure, i.e. organization of the mental lexicon (de Deyne et al. 2016).[4] Clearly, a lot has happened during the last two decades, yet more can be done especially with respect to indexing (the organization of the data) and navigation.

This paper is organized as follows. I start by providing evidence that storage does not guarantee access. That this holds for humans has been shown already 50 years ago (Tulving and Pearlstone 1966), in particular via Brown and Mc Neill’s (1966) seminal work devoted to the *tip-of-the-tongue problem*. I will show here that this can also hold for machines. The assumption that what is stored can also be accessed (anytime), is simply wrong. To illustrate this claim I will compare an extended version of WN (Mihalcea and Moldovan 2001) to an equivalent resource based on Wikipedia

[2] More or less well specified thoughts (concepts, elements of the word’s definition), or somehow related elements: collocations, i.e. associations (elephant: tusk, trunk, Africa).

[3] https://www.onelook.com

[4] For a short survey of some of this work see Zock et al. (2009), and Zock (2015a, b).

(WiPe). Next, I will discuss under what conditions WN is adequate for word access, and finally, I will sketch a roadmap describing the steps to be performed in order to go beyond this very popular resource. The goal is to build a navigational tool (index, association network) allowing authors to go from the word they know (word available when being in the ToT state) to the word they are looking for (target). Before doing so, I will present though my theory concerning the dialogue between the dictionary user and the lexical resource.

14.2 Storage Does Not Guarantee Access

To test this claim let me describe here briefly an experiment carried out with a colleague of mine (Zock and Schwab 2011). We ran a small experiment, comparing an extended version of WN (henceforth, WN-x) 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 as to show, firstly, that storage does not guarantee access and, secondly, that access depends on a number of factors like (a) quality of the resource within which the search takes place (organisation, completeness), (b) index, and (c) type of the query (proximity to the target).[5] Having two resources built with different foci, our goal was to check the efficiency of each one of them with respect to word access. For practical reasons we considered only direct neighbors. Hence, we defined a function called *direct neighborhood*, which, once applied to a given window (sentence/paragraph,[6] produces all its co-occurences. Of course, what holds for *direct associations* (our case here), holds also for indirectly related words, that is, words whose distance >1 (mediated associations).

14.3 Comparisons of the Two Resources

Table 14.1 shows the results produced by WN-x and WiPe for the following, randomly given 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

[5] To show the relative efficiency of a query, D. Schwab has developed a website in Java as a servlet. Usage is quite straightforward: people add or delete a word from the current list, and the system produces some output. The output is an ordered list of words, whose order depends on the overall score [i.e. the number of co-occurrences between the input, i.e. ‘source word’ (S_W) and the directly associated words, called ‘potential target word’ (PT_W)]. For example, if the S_W ‘bunch’ co-occured 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.

[6] Optimal size is an empirical question, which may vary with the text type (encyclopedia vs. raw text).

Table 14.1 Comparing two corpora with various inputs

Input	Output: WN-x	Output: WiPe
wine	**488 hits** grape, sweet, serve, France, small, fruit, dry, bottle, produce, red, bread, hold…	**3045 hits** name, lord characteristics, christian, grape, France, … vintage (81st), …
harvest	**30 hits** month, fish, grape, revolutionary, calendar, festival, butterfish, dollar, person, make, wine, first, …	**4583 hits** agriculture, spirituality, liberate, production, producing, …, vintage (112th), …
wine + harvest	**6 hits** make, grape, fish, someone, commemorate, person, …	**353 hits** grape, France, vintage (3rd), …

of them (6 hits). What is more interesting though is the fact that none of these terms allows us to access the target, eventhough it is contained in the database of WN-x, which clearly supports our claim that storage does not guarantee access. Things are quite different for an index built on the basis of information contained in WiPe. 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 not bad at all. I 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 an index allowing users to find the elusive word.

One may wonder why we failed to access information contained in WN and why WiPe performed so much better. We believe that the relative failure of WN is mainly due to the following two facts: the size of the corpus (114,000 words as opposed to 3,550,000 for WiPe), and the number of syntagmatic links, both of which are fairly small compared to WiPe. Obviously, being an encyclopedia, WiPe contains many more *syntagmatic* links than WN. Of course, one could object that we did not use the latest release of WN which contains many more words (147,278 words, clustered into 117,659 synsets). True as it is, this would nevertheless not affect our line of reasoning or our conclusion. Even in this larger resource we may fail to find what we are looking for because of the lack of *syntagmatic links*.[7]

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). Yet, in order to be fair

[7] It should be noted though that serious efforts have been made to enrich WN by adding syntagmatic links (Bentivogli and Pianta 2004) and various kinds of encyclopedic information: topic signatures (Agirre et al. 2001), domain-specific information (Boyd-Graber et al. 2006; Gliozzo and Strapparava 2008; Fernando 2013), etc., but none of them seems to be integrated in the version accessible via the web interface (http://wordnetweb.princeton.edu/perl/webwn). Yet this is the one accessed by the ordinary language user who is generally either not able or willing to spend time to write an algorithm to integrate the different resources.

Table 14.2 Comparing two corpora with various inputs

	Output: WN-x	Output: WiPe
ball	**346 hits** game, racket, player, court, volley, Wimbledon, championships, inflammation, …, tennis (15th), …	**4891 words** sport, league, football, hand, food, foot, win, run, game, …, tennis (27th), …
racket	**114 hits** break, headquarter, gangster, lieutenant, rival, kill, die, ambush, tennis (38th), …	**2543 words** death, kill, illegal, business, corrupt, …, tennis (72nd), …
ball + racket	**11 hits** game, tennis, (2nd), …	**528 hits** sport, strike, tennis (3rd), …

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 get 'vintage', as the term 'wine' is contained in its definition ('vintage': a season's yield of 'wine' from a vineyard). Note that in such cases even Google works often quite well, but see also (Bilac et al. 2004; El-Kahlout and Oflazer 2004; Dutoit and Nugues 2002).

Last but not least, success may vary quite dramatically, depending on the input (quality of the query). As you can see in Table 14.2, WN performs slightly better than WiPe for the words 'ball', 'racket' and 'tennis'. Yet, WiPe does not lag much behind; additionally, it contains many other words possibly leading to the target words ("player, racket, court", ranked, respectively in position 12, 18 and 20).

Not being an encyclopedia, WN lacks most of them, though surprisingly, it contains named entities like 'Seles' and 'Graf', two great female tennis players of the past. Given the respective qualities of WN and WiPe one may well consider integrating the two by relying on a resource like *BabelNet* (Navigli and Ponzetto 2012).[8] This could be done in the future. In the meantime let us take a closer look at WN and its qualities with respect to word look up.

14.4 Under What Condition Is WN Really Good for Consultation?

It is a well-kown fact that WN is based on psycholinguistic principles (associations, network, hierarchical structure, …). What is less known though is the fact, that despite its origines, it has never been built for consultation. It has been primarily conceived for usage by machines: "WordNet is an online lexical database designed for use under

[8] http://lcl.uniroma1.it/babelnet/.

program control." (Miller 1995, p. 39). This being said, WN can nevertheless be used for consultation, all the more as it is quite good at it under certain circumstances.

Remains the question under what conditions WN is able to reveal the elusive target word. I believe that it can do so perfectly well provided that the following three conditions are met:

(a) the *author knows* the *link* holding between the source word (input, say 'dog') and the target, e.g. ([dog] + <u>*synonym*</u> = [? target]) → ([target = bitch]); ([dog] + <u>*hypernym*</u> = [? target]) → ([target = canine]);
(b) the *input* (source word) and the *target* are *direct neighbors* in the resource. For example, [seat]-[leg] (*meronym*); or [talk]-[whisper] (*troponym*), …
(c) the *link* is *part* of WN's database, e.g. 'hyponym/hypernym', 'meronym', …

14.5 The Framework of a Navigational Tool for the Dictionary of the Future

To access a word means basically to reduce the entire set of words stored in the resource (lexicon), to one (target). Obviously, this kind of reduction should be performed quickly and naturally, requiring as little time and effort (minimal number of steps) as possible on the users' side. Note that this process is knowledge based, meaning that the user may have stored not only the elusive word but also other, somehow related words. This is a very important point, as in case of failure, the dictionary user may well start from any of these connected words.

When I wrote that WN is quite successful with regard to word look-up under certain circumstances, I also implied that it is not so good when these conditions are not met. More precisely, this is likely to occur when:

(a) the source (input) and the target are only *indirectly* related, the distance between the two being greater than 1. This would be the case when the target ('Steffi Graf') cannot be found directly in reponse to some input ('tennis player'), but only via an additional step, say, 'tennis pro'—([tennis player] → [tennis pro])—given as input at the next cycle, in which case it will at best only then reveal the target.[9]
(b) the input ('play') and the target ('tennis') belong to different parts of speech (see 'tennis problem', Fellbaum 1998);
(c) the prime and the target are linked via a *syntagmatic association* ('smoke'-'cigar'). Since the majority of relations used by WN connect words from the

[9] Note that the situation described is a potential problem for any association network. Note also that, eventhough Named Entities (NEs) are generally not contained in a lexicon, some of them have made it into WN. This is the case for some famous tennis players, like Steffi Graf. Anyhow, since NEs are also words, the point we are trying to make holds for both. Hence, both can be organized as networks, and whether access is direct or indirect depends on the relative proximity of the input (prime) with respect to the target word.

same part of speech, word access is difficult if the output (target) belongs to a different part of speech than the input (prime)[10];

(d) the user ignores the link, he cannot name it, or the link is not part of WN's repertory.[11] Actually this holds true (at least) for nearly all syntagmatic associations.

Let us see how to go beyond this. To this end I present here briefly the principles of the resource within which search takes place, as well as the required navigational aid (categorial tree) to allow authors to find quickly the word they are looking for. Yet, before doing so, let me clarify some differences between hierarchically structured dictionaries and my approach.

While lexical ontologists (LO) try to integrate all words of a language into a neat subsumption hierarchy, we try to group them mainly in terms of direct neighborhood. More precisely, we try to build a lexical graph where all words are connected, regardless of whether we can name the link or not. Put differently, we try to build a hybrid association network whose elements (words) are connected via typed and untyped links. Both kinds of links are necessary for filtering, i.e. to ensure that the search space is neither too big (typed links), nor too small (untyped links). Knowledge of the relationship between the source and the target is an obvious asset, as it reduces considerably the search space. Yet, untyped links are a necessary evil: they address the fact that two words evoke each other. Hence, even if we cannot name the link, we should still include the connected word in the list within which search takes place. Otherwise, how can the user find it? Of course, untyped links can cause growth of the search space. Yet, in order to avoid this problem we could group by category the words devoid of a link (Fig. 14.1, step 2). Obviously, this approach yields a quite different network than WN. Hence it will also produce different results than WN for a given input (see Table 14.3).

Suppose we started from a broad term like 'food'. A lexical ontology like WN would produce the entire list of objects referring to 'food' (hyponyms), while an association network would only reveal typically evoked words {food, bread, noodles, rice, fish, meat, cook, eat, buy, starving, good, expensive, fork, chopsticks….}. This list contains, of course, a subset of the terms found in a LO (terms referring to 'food'), but it also contains syntagmatically related words (*origine*: France; *state*: hungry, …). Compare the respective results obtained by WN and the Edinburgh Association Thesaurus.[12]

By taking a look at this second list one can see that it contains not only hyponyms, that is, specific kinds of food (meat, cheese, …), but also syntagmatically related words (cook, good, France, …), i.e. words typically co-occurring with the term 'food'. Note that our list may lack items like 'bagles', 'cheese' or 'olives'. This is quite normal, if ever these words are not strongly associated with our input (food), which does not imply, of course, that we cannot activate or find them. Had we given

[10] This being said, WN does have cross-POS relations, i.e. "morphosemantic" links holding among semantically similar words: observe (V), observant (Adj) observation (N).

[11] For example: 'well-known_for', 'winner_of', …

[12] http://www.eat.rl.ac.uk (see also: http://rali.iro.umontreal.ca/word-associations/query/).

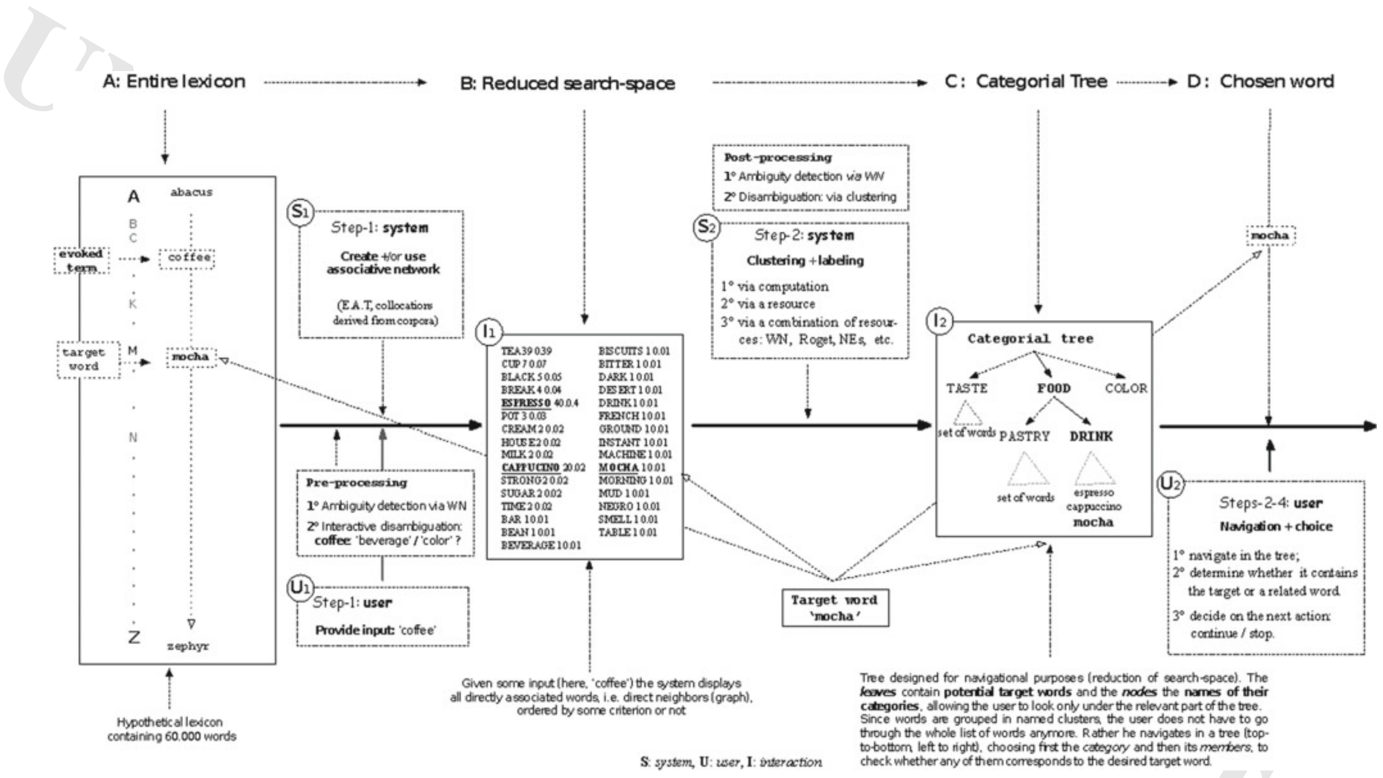
A: Entire lexicon
B: Reduced search-space
C: Categorial Tree
D: Chosen word
A
abacus
B
C
evoked term
coffee
K
target word
M
mocha
N
Z
zephyr
Hypothetical lexicon containing 60.000 words
S1
Step-1: system
Create +/or use associative network
(E.A.T, collocations derived from corpora)
Pre-processing
1° Ambiguity detection via WN
2° Interactive disambiguation: coffee: 'beverage' / 'color' ?
U1
Step-1: user
Provide input: 'coffee'
I1
TEA39 0.39
CUP 7 0.07
BLACK 5 0.05
BREAK 4 0.04
ESPRESSO 40.0.4
POT 3 0.03
CREAM 2 0.02
HOUSE 2 0.02
MILK 2 0.02
CAPPUCINO 20.02
STRONG 2 0.02
SUGAR 2 0.02
TIME 2 0.02
BAR 10.01
BEAN 1 0.01
BEVERAGE 10.01
BISCUITS 1 0.01
BITTER 1 0.01
DARK 1 0.01
DESERT 1 0.01
DRINK 1 0.01
FRENCH 10.01
GROUND 10.01
INSTANT 10.01
MACHINE 1 0.01
MOCHA 10.01
MORNING 1 0.01
MUD 1 0.01
NEGRO 1 0.01
SMELL 1 0.01
TABLE 1 0.01
Given some input (here, 'coffee') the system displays all directly associated words, i.e. direct neighbors (graph), ordered by some criterion or not
Post-processing
1° Ambiguity detection via WN
2° Disambiguation: via clustering
S2
Step-2: system
Clustering + labeling
1° via computation
2° via a resource
3° via a combination of resources: WN, Roget, NEs, etc.
Target word 'mocha'
S: system, U: user, I: interaction
I2
Categorial tree
TASTE
FOOD
COLOR
set of words
PASTRY
DRINK
set of words
espresso
cappuccino
mocha
Tree designed for navigational purposes (reduction of search-space). The leaves contain potential target words and the nodes the names of their categories, allowing the user to look only under the relevant part of the tree. Since words are grouped in named clusters, the user does not have to go through the whole list of words anymore. Rather he navigates in a tree (top-to-bottom, left to right), choosing first the category and then its members, to check whether any of them corresponds to the desired target word.
mocha
U2
Steps-2-4: user
Navigation + choice
1° navigate in the tree;
2° determine whether it contains the target or a related word.
3° decide on the next action: continue / stop.

Fig. 14.1 Lexical access a four-step process for the user

Table 14.3 The respective outputs produced by a *lexical ontology* (here WN) as opposed to an *association network*, here, the Edinburgh Association Thesaurus (E.A.T)

WN: *hypernym*: solid; *part_holonym*: nutrient; **hyponyms**: leftovers, fresh_food, convenience_food, chocolate, baked_goods, loaf, meat, pasta, health_ food, junk_food, breakfast_food, green_goods, green_groceries, coconut, coconut_meat, dika_bread, fish, seafood, butter, yoghourt, cheese, slop
E.A.T: at, drink, good, thought, dinner, eating, hunger, salad, again, apple, baby, bacon, bread, breakfast, case, cheese, consumption, cook, firm, fish, France, goo, great, hungry, indian, kitchen, lamb, loot, meal, meat, mix, mouth, noah, nosy, of, pig, please, poison, rotten, sausage, steak, stomach, storage, store, stuff, time, water, yoghurt, yum

'wine' or 'oil' 'green' and 'Greece' as input, chances are that 'cheese' and 'olives' would pop up immediately, while they are burried deep down in the long list of food produced by a LO.

Let us return to the problem of word access. Just as orientation in real world requires tools (map, compass) we need something equivalent for locating a word in a lexical resource. While the *semantic map* defines the territory within which search takes place, the *lexical compass* guides the user, helping her to reach the goal (target word). Obviously, the terms map and compass are but metaphors, as there are important differences between world maps and lexical graphs (see below) on one hand, and compasses sailors use and the tool an information seeker is relying on (human brain) on the other. The map I have in mind is basically an association network. It is a fully connected graph encoding all directly associated words given some input. This kind of graph has many redundancies, and the links are not necessarily labeled. In this respect it is very different from WN and even more so from the maps we use when traveling in real world. Also, when using a world map the user generally knows more or less precisely the destination or the relative location of the place he is looking for, for example, south of Florence. He may also be able to deduce its approximate location, eventhough she is not able to produce its name (Rome). This does not hold in the case of a user resorting to a lexical resource (map) based on associations. While the user may know the starting point (knowledge available when trying to find the target, the elusive word), he cannot name the destination (target), as if he could, there would be no search problem to begin with. The user is either able to activate the word (in which case the problem is solved), or not. In this latter case all he can do is to rely on available knowledge concerning the target, an assumption I clearly make here. For example, users often know a related word, and they know how it relates to the target: (part of the) meaning, sound, collocational, etc. Knowledge is often fragmentary. Yet, incomplete as it may be, this kind of information may allow us to help them to find the target, guiding him in a reduced, clearly marked search space (details here below).

To get back to navigation in real world. In the case of spatial navigation it suffices to know that 'Rome' is south of 'Florence', which is part of 'Lazio', and that it can be reached by car in about 2 h. Having this kind of knowledge we could initiate search in the area of 'Lazio', since 'Lazio' is an area south of 'Tuscany', the area containing

‘Florence’. While this strategy works fine in the case of spatial navigation, it will not work with lexical graphs. In this kind of network terms are related in many ways and their strength may vary considerably. Hence, it is reasonable to show a term only if it is above a certain threshold. For example, a term A (Espresso) being connected to term B (coffee) may be shown only if it is sufficiently often evoked by B. Note that eventhough words are organized in terms of neighborhood, the link between them (explicited or not) may be of many other kinds than a spatial relation. In sum, the links connecting words in an associative network are much more diverse than the ones typically found in a lexical ontology.

As mentioned already, humans using world maps usually know the name of their destination, whereas people being in the ToT state do not. Yet, even if they did, they would not be able to locate it on the map. Lexical graphs are simply too big to be shown entirely on a small screen.[13] In sum, we need a different approach: search must be performed stepwise, taking place in a very confined space, composed of the input and the direct neighbors (directly associated words). It is like a small window moved by the user from one part of the graph to the next. If there are differences between world maps and association networks (lexical graphs), there are also 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, target), the latter (brain) assumes the user to know, the goal, i.e. target word,[14] or its direction (even if one does not know its precise location). While the user cannot name the goal—he has only *passive knowledge* of it,—the system cannot guess it. However it can make valuable suggestions. In other words, eventhough the system can only make suggestions concerning the target or the directions to go (which word to use as input for the next cycle), it is the user who finally decides whether the list contains the target or not, and if so, in what direction to go. He is the only one to know which suggestion corresponds best to the target (the word he has in mind) or which one of them is the most closely connected to it. Of course, the user may go wrong, but as experience shows his intuitions are generally quite good.

Before sketching a roadmap concerning the scenario of word access via the still-to-be-built resource (association network), let me quickly provide some background information concerning the users’ knowledge, a critical component in this kind of dialogue.

[13] Associative networks contain many redundancies and are potentially endless, since they contain loops. For example, an input, say ‘Rome’ may well appear to be the direct neighbor of one of its outputs, ‘Italy’: ([Rome] → {[capital], [Italy], [city]}); ([Italy] → {[country], [France], [Rome]}).

[14] It has been shown over and over again that people being in the ToT state are able to identify immediately, and without making any mistakes the target word if it is shown to them, eventhough they could not name it. This is passive knowledge.

14.6 Navigation, a Fundamentally Cognitive Process

As I will show in this section, navigation in a lexical resource is above all a knowledge-based process. Before being able to access a word, we must have acquired it. It is only then that it has become part of our knowledge. Yet, storage does not guarantee access (Zock and Schwab 2011). This fact has not received the attention it deserves by lexicographers. Note also that there are several kinds of knowledge: declarative, meta-knowledge (not necessarily linguistic) and knowledge states.

- *Declarative knowledge* is what we acquire when learning words (meaning, form, spelling, usage), and this is the information generally encoded in dictionaries. Obviously, in order to find a word or to find the information associated with it, they must be stored, though this is not enough.
- Next, there is *meta-knowledge*, which also needs to be acquired. Being generally unavailable for in(tro)spection, meta-knowledge reveals itself in various ways. For example, via the information available when we *fail to access* a *word* (Schwartz 2006), or via the *query* we provide at the moment of launching a search. As word association experiments have shown (Aitchison 2003) words always evoke something. Since this is true for all words one can conclude that all words are connected in our mind, which implies that all words are accessible from anywhere like in a fully connected graph.[15] All we have to do is to provide some input (source word, available information) and follow then the path linking this input to the output (target). Interestingly, people hardly ever start from words remotely related to the target. Quite to the contrary, the words they give at the input (source words) tend to be more or less direct neighbors of the target, requiring generally only one or two steps for the solution, that is, they are hardly ever further away than the distance of two (steps).[16] Also, dictionary users often know the type of relationship holding between the input (prime) and the target, otherwise, why would lexicographers build thesauri, synonym-or collocation dictionaries? All these observations lend support to our intuition that people have a considerable amount of (meta-) knowledge concerning the organization of words in their mind, i.e. their mental lexicon.
  The notion of relationship has been nicely exploited by WN, which due to this feature keeps the search space, i.e. a set of candidates among which the user has to choose, quite small. The idea of relatedness has led lexicographers to build thesauri, collocation- and synonym dictionaries. Obviously an input consisting only of a

[15] Note that this does not hold for WN, as WN is not a single network, but a set of networks. There are 25 for nouns, and at least one for all the other parts of speech.

[16] This is probably one of the reasons why we would feel estranged if someone provided as cue 'computer', while his target were 'mocha'. The two are definitely not directly connected, though, there is a path between them, eventhough it is not obvious (The chosen elements are always underlined.): *computer* → (*Java, Perl, Prolog; mouse*, *printer; Mac*, *PC*); (1) Java → (*island, programming language*); (2) Java (island) → (coffee; Kawa Igen); (3) coffee → (cappucino, mocha, latte). Note that 'Java' could activate 'Java beans', a notion inherent to JAVA, the programming language. In this case it would lead the user directly to the class (hypernym) containing the desired target word (mocha).

simple word is hard to interpret. Is the user looking for a more general/specific word, a synonym or antonym? Is the input semantically or phonetically related to the target, or is it part of the target word's definition (dog-animal)? In each case the user is expecting a different word (or set of words) as output. Hence, in order to enable a system to properly interpret the users' goals we need this kind of metalinguistic information (neighbor of the target, i.e. source word + relation to the target) at the input.[17] If ever the user cannot provide it, the system is condemned to make a rough guess, presenting all directly connected words. Obviously, such a list can become quite large. This being so, it makes sense to provide the system this kind of information to produce the right set of words, while keeping the search space small.

- *Knowledge states*, refer to the knowledge activated at a given point in time, for example, when launching a search. What has been primed? What is available in the user's mind? Not all information stored in our mind is equally available or prominent anytime. The fact that peoples' *knowledge states* vary is important, as it co-determines the way a user proceeds in order to find the information he is looking for. This being so, it is important to be taken into consideration by the system designer. In conclusion, all this knowledge must be taken into account as it allows us to determine the search space, reducing its scope, which otherwise is the entire lexicon.

The example here below illustrates to some extent these facts with regard to wordfinding in an electronic resource. Suppose you are looking for a word conveying the idea of a *large black-and-white herbivorous mammal of China*. Yet, for some reason you fail to retrieve the intended form *panda*, even though you do know a lot concerning the target. People being in this state, called the ToT-problem, would definitely appreciate if the information they are able to access could be used to help them find the target. Figure 14.1 illustrates the process of getting from a visual stimulus to its corresponding linguistic output (word, expression) via a lexical resource. Given an external stimulus (A) our brain activates a set of features (B) that ideally allow us to retrieve the target form. If our brain fails, we use a fallback strategy and give part of the activated information to a lexical resource (C) expecting it to filter its base (D) in the hope to find the target (panda) or a somehow related word (E). As one can see, we consider look-up basically as a two-step process. At step one the user provides some input (current knowledge) to which the system answers with a set of candidates, at step two the user scans this list to make her choice (Table 14.4).

[17] This has of course consequences with respect to the resource. To be able to satisfy the different user needs (goals, stratgies) we probably need to create different databases: Obviously, to find a target on the basis of sound (rhymes), meanings (meaning-fragments) or related words (co-occurrences), requires networks encoding a different kind of information.

Table 14.4 Lexical access a two-step process mediated by the brain and an external resource (lexicon)

A: Perceptual input, i.e. target	B: Associated features in the mental lexicon (brain)	C: Input to lexical resource	D: Lexical resource	E: Output of lexical resource
	Type: bear *Lives_in*: China *Features*: black patches *Diet*: eats bamboo	bear China	**a**ardvark … … *panda* … … theorem … **z**ygote	*panda* polar bear

14.7 The Roadmap

Since alphabetically organized dictionaries are not very useful for onomasiological search (language production), we follow WN by organizing words in terms of neighborhood. All words are connected, and if possible, the links are given names. This being so, we have a map supporting navigation. The user can enter the graph at any point, to follow the links until having reached the target. Obviously, the kind of links, as well as the presence/absence of link names (see below) are important with respect to the search space (see below). A network devoid of link names yields many more hits for a given input than a network containing the same set of words but whose links are named. Imagine the number of possible outputs for [‘tree’] compared to [(‘tree’) + (‘synonym’/‘hypernym’].

Let us now see quickly how to make all this work. Imagine an author wishing to convey the name of a beverage commonly found in coffee shops (target: ‘mocha’). Failing to do so, he reaches for a lexicon. Since dictionaries are too huge to be scanned from cover (letter A) to cover (Z), I suggest to create a dialog between the user and the computer to reduce incrementally the search space. The user provides the input,[18]—word coming to his/her mind (source) when trying to access the target,—and the system produces as output a list of potential target words, in the hope to enable the user to find the elusive word.

Note that, concerning the source and the target, there are basically three cases: (A) the two are directly related, and the user knows their relationship; (B) the user knows a direct neighbor, but he ignores the name of their relationship; (C) the source and the target are not directly related, they are only indirect neighbors. Since the first case (A) is quite well handled by WN, I will illustrate here only ‘B’, ignoring the last case (C), as it can be solved indirectly by applying recursively the procedures

[18] This latter can be a single word—‘coffee’ in the case of target ‘mocha’—or a set of words, which in a normal communicative setting would yield a sentence, where the information seeker asks someone else to help him to find the elusive word.

proposed in ‘A’ and ‘B’. [19] To convey as simply as possible the rationale underlying my approach let us make the following assumptions: (A) the user’s input is a single word, here ‘coffee’ (step-1, Fig. 14.1); (B) the target, i.e. the elusive form, is ‘mocha’; (C) the two are directly related in the resource in which search takes places; (D) the nature of their relationship has not been specified by the user.[20] Because of this last point—the relationship between the source and the target not being given with the input—search space may grow considerably. To avoid this problem I propose to present in a clustered and labeled form (categorial tree) all direct associates of the input (step-2, Fig. 14.1).[21]

The user navigates in this tree, deciding on the category within which to look for the target, and if he cannot find it in any of them, in what direction to go. If he could find the target, search stops, otherwise the user will pick one of the associated terms or provide an entirely new word, and the whole process iterates. The system will then respond with a new set of proposals.

Two points, one concerning step-1, the other step-2. Ideally, the searchspace determined at step-1 should contain the target word. As we have seen in the experiment described here above (Sect. 14.3), WN failed to reveal a candidate though it was stored it in its database. This failure was due to a lack of syntagmatic associations, that is, it was a side-effect of a design choice of how to connect words, or, which words to connect. Hence, the search space proposed by WN in response to some input was too small. At least it did not contain the target while its competitor (WiPe) did.

The categories of our tree (step-2) resemble somehow those of Roget’s thesaurus. Actually, they are not quite the same, but this is not really the point. What I would like to stress is the fact that both the categories and the words composing the search-space are dynamically computed in our case, while they are frozen, i.e. determined once and for all in Roget. Hence, the set of words (search space) to be presented in the categorial tree will be considerably smaller in our case than the ones displayed by Roget. I believe that this kind of flexibility is a desirable feature as we cannot predict the user’s input or his goals, we can only comply with them.

[19] This kind of wording can be generalized to a pattern for asking the following question: “What is the word for ‘[X] that [Y]?”, where [X] is usually a hypernym and [Y] a stereotypical, possibly partial functional/relational/case description (action) of the target word. A similar pattern could be used for namefinding. For example, asking “What is the name of the <conqueror> of <empire>?” could yield ‘Pizarro’ or ‘Cortés’, depending on the value of the empire (Inca/Aztec). As one can see, the processes underlying wordfinding and namefinding are not very different.

[20] Note, that in order to determine properly the initial search space (step-1), we must have already well understood the input [mouse$_1$/mouse$_2$ (rodent/device)], as otherwise our list will contain a lot of noise, presenting ‘cat, cheese’ together with ‘computer, mouse pad’ {cat, cheese, computer, mouse pad}, which is not quite what we want, since some of these candidates are irrelevant, i.e. beyond the scope of the user’s goal.

[21] This labeling is obligatory to allow for realistic navigation, as the list produced in response to the input may be very long and the words being of the same kind may be far apart from each other in the list. Hence it makes sense to structure words into groups by giving them appropriate (i.e. understandable) names so that the user, rather than looking up the entire list of words, searches only within a specific bag labeled by a category.

To get back to our roadmap. As one can see, the proposed method is quite straightforward, reducing considerably time and space needed for navigation and search. Suppose that you had to locate a word in a resource of 50,000 words. If your input triggered 100 direct associates, one of them being the target, then we would have reduced in a single step the search space by 99.8%, limiting navigation and search to a very small list. Suppose that our hundred words were evenly spread over 5 groups, than search would consist in spotting the target in a list of 25 items: 5 being category names and 20 being words within the chosen group.

A small note concerning the 2nd step. Step-2 yields a tree whose leaves are *potential target words* and whose nodes are *categories*, which, while being also words, are not at all the goal of the search. They are only the means to reach the goal. Put differently, their function is orientational, guide the user during his search.

Words at the leave-level are potential target words, while the ones at the intermediate level (category names; preterminal nodes) are meant to reduce the number of words among which to perform search, and to help the user to decide on the direction to go. Hence, category names are reductionist and orientational (signposts), grouping terminal nodes into a bag, signaling via their name not only the bag's content, but also the direction to go. While the system knows the content of a bag, it is only the user who can decide which of the bags is likely to contain the elusive word. Because, eventhough he cannot name the target, he is the only one to know the target, be it only passively and in fairly abstract terms. This is where the categoy names have their role to play. In sum, it is not the system that decides on the direction to go next, but the user. Seeing the names of the categories she can make reasonable guesses concerning their content.

In conclusion, categories act somehow like signposts signaling the user the kind of words he is likely to find choosing one bag rather than another. Indeed, knowing the name of a category (fruit, animal), the user can guess the kind of words contained in each bag (kiwi vs. crocodile). Assuming that the user knows the category of the searched word,[22] she should be able to look in the right bag and take the best turn. Navigating in a categorial tree, the user can search at a fairly high level (class) rather than at the level of words (instances). This reduces not only the cognitive load, but it increases also chances of finding the target, while speeding up search, i.e. the time needed to find a word.

While step-1 is mainly a matter of *relatedness* ('wine' and 'red' being different in nature, they are nevertheless somehow related), step-2 deals with *similarity*: there are more commonalities between 'dogs' and 'cats' than between 'dogs' and 'trees'. Put differently, the first two terms are more similar in kind than the last two. The solution of the second step is certainly more of a challenge than the one of step-1 which is largely solved, eventhough there is an issue of relevance: not all co-occurences

[22] A fact which has been systematically observed for people being in the ToT state who may tell the listener that they are looking for the name of a "fruit typically found in a <PLACE>", say, New Zealand, in order to get 'kiwi'.

are really useful.[23] To put words into clusters is one thing, to give them names an ordinary dictionary user can understand is quite another.[24] Yet, arguably building this categorial tree is a crucial step, as it allows the user to navigate on this basis. Of course, one could question the very need of labels, and perhaps this is not too much of an issue if we have only say, 3–4 categories. We are nevertheless strongly convinced that the problem is real, as soon as the number of categories (hence the words to be classified) grows.

To conclude, I believe it is fair to say that the 1st stage seems to within reach, while the automatic construction of the categorical tree remains a true challenge despite some existing tools (word2vec) and the vast literature devoted to this specific or otherwise strongly related problems (Zhang et al. 2012; Biemann 2012; Everitt et al. 2011).

One last point: to be truly useful, the user should provide as input not only a word, but also a clue concerning the relationship between this input and his goal (target word). Does he look for a semantically, formally (sound) or otherwise related word with respect to the input? Since inputs can be interpreted in many ways, we need additional information. Given some input, what is the user's goal? Is he looking for a synonym, hypernym or a similarly sounding word? Obviously, different goals yield different searchspaces. This is a very important point. Authors searching for a sound-related word to 'right' expect a different set of candidates (write, wright, rite), from authors looking for its antonym ('wrong'). WN takes advantage of this fact eventhough only a subset of the ones mentioned here below are actually implemented. Table 14.5 shows some possible links between some input (prime) and its directly associated output (possible target).

Note that these links are of different sort: some are conceptual (1–13), others are formal, i.e. they concern linguistic forms (15–19), and what we dubbed 'free association' (14) concerns both. For more details concerning links or relations, take a look at (Evens 2009; Green et al. 2002; Miller and Fellbaum 1992; Murphy 2003; Nastase et al. 2013; Storjohann 2010).

14.8 Conclusion

Obviously, the human brain is a complex object and so is the process of accessing words in the mental lexicon. My goal was not so much to address the problem of complexity, i.e. the topology of the map of the mental lexicon. My goal was rather to describe a method, that, once implemented, should help people to overcome the

[23] Take for example the Wikipedia page devoted to 'panda', and check which of the co-occurrences are those typically evoked when looking for this particular lexical concept.

[24] For example, while the sequence of hypernyms listed by WN for *horse* captures much of the phylogenetic detail a biologist would want to see recorded (horse → equine → odd-toed ungulate → ungulate → placental mammal → mammal → vertebrate → chordate → animal → organism → entity), most of these terms mean next to nothing to an ordinary dictionary user.

Table 14.5 Possible links or associations between an input (cue/prime) and the target

	Type of relation	Description of the relation	Prime-target
1	*Hypernym*	A more *general* word	pie-pastry
2	*Hyponym*	A more *specific* word	fruit-*nut*
3a	*Meronym*_substance	A concept being a *substance* of another concept	*blood*-body
3b	*Meronym*_part_of	A concept being *part* of another concept	*ship*-fleet
3c	*Meronym*_member_of	A concept being a *member* of another concept	*kid*-family
4a	*Holonym*_substance	A concept having another concept as *substance*	sea-*salt*
4b	*Holonym*_part_of	A concept having another concept as *part*	tree-*leave*
4c	*Holonym*_member_of	A concept having another concept as *member*	team-*player*
5	*Cause* to	A verb expressing the *cause* of a result	kill-*die*
6	*Entailment*	A verb expressing an unavoidable *result*	buy-*have*
7	*Troponym*	A specific *way* to perform an action	drink-*sip*
8	*Part_of_meaning*	Part of the target word's definition	butter-milk
9	*Quality*	Typical quality, or inherent feature	snow-*cold*
10	*Co-occurrence*	Two concepts occurring frequently together	blue-*sky*
11	*Topically related*	Two concepts related by topic	sea-tide
12	*Used_for*	Instrumentally related words	fork-eating
13	*Made_of*	Substance or element used to make <object>	glass-sand
14	*Free association*	Can be any kind of link between two words	door-open
15	*Synonym*	Word expressing basically the *same* meaning	cup-mug
16	*Antonym*	A word meaning the *opposite*	dry-wet
17	*Sound (rhyme)*	Two similar sounding words	bad-mad/sad
18	*Homophones*	Words sounding alike, but spelled differently	right-write
19	*Anagrams*	Composed of same or similar components	cheater-teacher

ToT-problem. The method is radically knowledge-based, that is to say, it takes into account knowledge users may have at the onset of consultation (see below).

I have started the paper by observing that word access remains a problem for dictionary builders (Thumb 2004) and users alike, in particular for those being in the production mode (Zock 2015a, b; Zock and Tesfaye 2015). Next I have shown that word storage does not guarantee its access, even if the target is stored in a computer. I have then analyzed some of the reasons why even a psycholinguistically motivated resource like WN often fails to reveal the word authors are looking for.

Finally, I have recasted the problem within a cognitive framework, presenting a roadmap of how to overcome the ToT-problem. The idea is to build a navigational tool (hybrid association network) to help humans to find the word they are looking for. The user provides the information coming to his mind when failing to access the target word (input), and the resource produces a list of potential target words (output). If the user can provide a (direct) neighbor of the target and its link, the answer is generally straightforward. In the opposite case I suggest to present the candidates in a labeled cluster-form (categorial tree) rather than as a huge, flat list. While the system‘s task is search-space reduction in step-1 (Fig. 14.1), its function in step-2 (building of the categorial tree) is to support navigation. Just as it is unreasonable to perform search in the entire lexicon, is it cumbersome to drill down huge lists. This is why I suggested to cluster and label the outputs produced in response to the query. After all, we want users to find the target quickly and naturally, rather than drown them under a huge, unstructured (or poorly structured) list of words.

Note that there is at least one study supporting the idea that knowledge of link names is useful. Nikolova et al. (2010) could show that word-finding is enhanced when wordnets contain syntagmatic links. They describe a study where people struck by aphasia used their resource, showing that retrieval was significantly better in this case than when relying on a resource devoid of this information. They conceded though that finding the first word to start communication with was still a problem. For other related work see (Ferret 2015; Zock and Biemann 2016).

One last point: the success of the (yet-to-be-built) resource hinges critically on three kinds of knowledge: (a) *factual knowledge*: to find a word it must exist, i.e. it must be stored; (b) *metaknowledge*: to allow for word access, words must also be well-organized, and the user must have some knowledge concerning this organization. This amounts to knowing at least some of the words connected to the target, and the relationship between some input (currently available word) and the goal (target word). Put differently, in order to be able to provide a decent input (typically a relatively close neighbor of the target), one must have at least a local view of the organization of the mental lexicon; (c) *cognitive states*: they are revealed by the word(s) coming to our mind when we search for a (word) form that we know, but cannot access. Solving this problem is what this paper has been about. So far this is only a concept, but I hope to be able to provide one evidence of its feasibility, as having such a tool would be extremely precious for dictionary users being in the production mode.

References

Abrams, L., Trunk, D. L., & Margolin, S. J. (2007). Resolving tip-of-the-tongue states in young and older adults: The role of phonology. In L. O. Randal (Ed.), *Aging and the elderly: Psychology, sociology, and health* (pp. 1–41). Hauppauge, NY: Nova Science Publishers Inc.

Abel, A. (2012). Dictionary writing systems and beyond. In S. Granger, & M. Paquot (Eds.), *Electronic lexicography* (pp. 83–106). Oxford: Oxford University Press.

Agirre, E., Ansa, O., Hovy, E., & Martinez, D. (2001). Enriching WordNet concepts with topic signatures. Retrieved from http://arxiv.org/abs/cs.CL/0109031.

Aitchison, J. (2003). *Words in the mind: An introduction to the mental lexicon*. Oxford: Blackwell.

Benson, M., Benson, E., & Ilson, R. (2010). *The BBI combinatory dictionary of English*. Philadelphia: John Benjamins.

Bentivogli, L., & Pianta, E. (2004). Extending WordNet with syntagmatic information. In P. Sojka, K. Pala, P. Smrz, C. Fellbaum, & P. Vossen (Eds.), *GlobalWor(l)dNet Conference, Proceedings* (pp. 47–53). Brno: Masaryk University.

Benjamin, M. (2014). Collaboration in the production of a massively multilingual lexicon. In *LREC Conference Proceedings* (pp. 211–215). Reykjavik.

Bernstein, T. (1975). *Bernstein's reverse dictionary*. New York: Crown.

Biemann, C. (2012). *Structure discovery in natural language*. Berlin: Springer.

Bilac, S., Watanabe, W., Hashimoto, T., Tokunaga, T., & Tanaka, H. (2004). Dictionary search based on the target word description. In *Proceedings of the Tenth Annual Meeting of The Association for Natural Language Processing* (pp. 556–559). Tokyo.

Boissière, P. (1862). *Dictionnaire analogique de la langue française: Répertoire complet des mots par les idées et des idées par les mots*. Paris: Auguste Boyer.

Boyd-Graber, J., Fellbaum, C., Osherson, D., & Schapire, R. (2006). Adding dense, weighted, connections to WordNet. In P. Sojka, Ks. Choi, C. Fellbaum, & P. Vossen (Eds.), *Proceedings of the Global WordNet Conference 2006* (pp. 29–35). Brno: Masaryk University.

Brown, A. S. (1991). The tip of the tongue experience: A review and evaluation. *Psychological Bulletin, 10,* 204–223.

Brown, R., & Mc Neill, D. (1966). The tip of the tongue phenomenon. *Journal of Verbal Learning and Verbal Behaviour, 5,* 325–337.

Casares, J. (1942). *Diccionario ideológico de la lengua española*. Barcelona: Gustavo Gili.

de Deyne, S., & Storms, G. (2015). Word associations. In J. R. Taylor (Ed.), *The Oxford handbook of the word*. Oxford, UK: Oxford University Press.

de Deyne, S., Verheyen, S., & Storms, G. (2016). Structure and organization of the mental lexicon: A network approach derived from syntactic dependency relations and word associations. In *Towards a theoretical framework for analyzing complex linguistic networks* (pp. 47–79). Berlin: Springer.

Dong, Z., & Dong, Q. (2006). *HowNet and the computation of meaning*. London: World Scientific.

Dornseiff, F., Wiegand, H. E., & Quasthoff, U. (2004). *Der deutsche Wortschatz nach Sachgruppen*. Berlin/New York: de Gruyter.

Dutoit, D., & Nugues, P. (2002). A lexical network and an algorithm to find words from definitions. In F. van Harmelen (Ed.), *Proceedings of the 15th European Conference on Artificial Intelligence* (pp. 450–454). Amsterdam: IOS Press.

Edmonds, D. (Ed.). (1999). *The Oxford reverse dictionary*. Oxford: Oxford University Press.

El-Kahlout, I. D., & Oflazer, K. (2004). Use of WordNet for retrieving words from their meanings. In P. Sojka, K. Pala, P. Smrž, C. Fellbaum, & P. Vossen (Eds.), *Proceedings of the Global Wordnet Conference* (pp. 118–123). Brno: Masaryk University.

Evens, M. W. (2009). *Relational models of the lexicon: Representing knowledge in semantic networks*. Cambridge: Cambridge University Press.

Everitt, B., Landau, S., Leese, M., & Stahl, D. (2011). Cluster analysis. Chichester: John Wiley & Sons, Ltd.

Fellbaum, C. (Ed.). (1998). *WordNet: An electronic lexical database and some of its applications*. Cambridge: MIT Press.

Fergusson, R., & Fergusson, R. (1985). *The Penguin rhyming dictionary*. London: Penguin.

Fernando, S. (2013). *Enriching lexical knowledge bases with encyclopedic relations* (Doctoral dissertation). University of Sheffield.

Ferret, O. (2015). Typing relations in distributional thesauri. In N. Gala, R. Rapp & G. Bel-Enguix, (Eds.), *Language Production, Cognition, and the Lexicon*. Springer, 113–134.

Fontenelle, T. (1997). *Turning a bilingual dictionary into a lexical-semantic database*. Tübingen: Max Niemeyer.

Gliozzo, A., & Strapparava, C. (2008). *Semantic domains in computational linguistics*. Berlin: Springer.

Granger, S., & Paquot, M. (Eds.). (2012). *Electronic lexicography*. Oxford: Oxford University Press.

Green, R., Bean, C. A., & Myaeng, S. H. (2002). *The semantics of relationships*. Dordrecht: Kluwer.

Hanks, P. (2012). Corpus evidence and electronic lexicography. In S. Granger & M. Paquot, (Eds.), *Electronic lexicography* (pp. 57–82). Oxford: Oxford University Press.

Kahn, J. (1989). *Reader's Digest reverse dictionary*. London: Reader's Digest.

Kilgarriff, A., Rychlý, P., Smrž, P., & Tugwell, D. (2004). *The sketch engine*. In G. Williams & S. Vessier (Eds.), *Proceedings of the Eleventh EURALEX International Congress* (pp. 105–116). Lorient, France: UBS.

Kilgarriff, A., & Kosem, I. (2012). Corpus tools for lexicographers. In S. Granger & M. Paquot (Eds.), *Electronic Lexicography* (pp. 31–56). Oxford: Oxford University Press.

Levelt, W., Roelofs, A., & Meyer, A. (1999). A theory of lexical access in speech production. *Behavioral and Brain Sciences, 22,* 1–75.

Levelt, W. J. (1993). *Speaking: From intention to articulation*. Cambridge: MIT Press.

Levelt, W. J. (2001). Spoken word production: A theory of lexical access. *Proceedings of the National Academy of Sciences, 98*(23), 13464–13471.

Mel’čuk, I., & Polguère, A. (2007). *Lexique actif du français: l’apprentissage du vocabulaire fondé sur 20 000 dérivations sémantiques et collocations du français*. Champs linguistiques. Bruxelles: De Boeck.

Mihalcea, R., & Moldovan, D. (2001). Extended WordNet: Progress report. In *NAACL 2001—Workshop on WordNet and Other Lexical Resources* (pp. 95–100). Pittsburgh, USA.

Miller, G. (1991). *The science of words. Scientific American Library*. New York: W H Freeman & Co.

Miller, G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM, 38*(11), 39–41.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography, 3*(4), 235–244.

Miller, G. A., & Fellbaum, C. (1992). Semantic networks of English. In B. Levin & S. Pinker (Eds.), *Lexical and conceptual semantics* (pp. 197–229). Cambridge and Oxford, England: Blackwell.

Moerdijk, F. (2008). Frames and semagrams. Meaning description in the General Dutch Dictionary. In *Proceedings of the Thirteenth Euralex International Congress* (pp. 561–570). Barcelona: *EURALEX*.

Murphy, M. L. (2003). *Semantic relations and the lexicon: Antonymy, synonymy and other paradigms*. Cambridge: Cambridge University Press.

Nastase, V., Nakov, P., Seaghdha, D. O., & Szpakowicz, S. (2013). Semantic relations between nominals. *Synthesis Lectures on Human Language Technologies, 6*(1), 1–119.

Navigli, R., & Ponzetto, S. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence, 193,* 217–250.

Nikolova, S., Tremaine, M., & Cook, P. R. (2010). Click on bake to get cookies: Guiding word-finding with semantic associations. In *Proceedings of the 12th International. ACM SIGACCESS Conference on Computers and Accessibility* (pp. 155–162). New York: ACM.

Rapp, B., & Goldrick, M. (2006). Speaking words: Contributions of cognitive neuropsychological research. *Cognitive Neuropsychology, 23*(1), 39–73.

Richardson, S., Dolan, W., & Vanderwende, L. (1998). Mindnet: Acquiring and structuring semantic information from text. In Proceedings of the 17th international conference on Computational linguistics, *ACL-COLING’98* (pp. 1098–1102). Montréal.

Robert, P., Rey, A., & Rey-Debove, J. (1993). *Dictionnaire alphabetique et analogique de la Langue Française*. Paris: Le Robert.

Roget, P. (1852). *Thesaurus of English words and phrases*. London: Longman.

Rundell, M., & Fox, G. (Eds.). (2002). *Macmillan English dictionary for advanced learners*. Oxford: Macmillan.

Schvaneveldt, R. (Ed.). (1989). *Pathfinder associative networks: Studies in knowledge organization*. Norwood, New Jersey, US: Ablex.

Schwartz, B. L. (2002). *Tip-of-the-tongue states: Phenomenology, mechanism, and lexical*. Mahwah, NJ: Lawrence Erlbaum Associates.

Schwartz, B. L. (2006) Tip-of-the-tongue states as metacognition. *Metacognition and Learning. 1*(2), 149–158.

Segalowitz, N. (2000). Automaticity and attentional skill in fluent performance. In H. Riggenbach (Ed.), *Perspectives on fluency* (pp. 200–219). Ann Arbor, MI: University of Michigan Press.

Sierra, G. (2000). The onomasiological dictionary: A gap in lexicography. In U. Heid, S. Evert, E. Lehmann, & C. Rohrer (Eds.), *Proceedings of the Ninth Euralex International Congress* (pp. 223–235). Stuttgart: IMS, Universität Stuttgart.

Storjohann, P. (Ed.). (2010). *Lexical-semantic relations: Theoretical and practical perspectives*. Amsterdam: John Benjamins Publishing.

Summers, D. (1993). *Language Activator: The world's first production dictionary*. London: Longman.

Thumb, J. (2004). *Dictionary look-up strategies and the bilingualised learner's dictionary. A think-aloud study*. Tübingen: Max Niemeyer Verlag.

Tulving, E., & Pearlstone, Z. (1966). Availability versus accessibility of information in memory for words. *Journal of Verbal Learning and Verbal Behavior, 5,* 381–391.

van Sterkenburg, P. (2003). Onomasiological specifications and a concise history of onomasiological dictionaries. In P. van Sterkenburg (Ed.), *A practical guide to lexicography* (pp. 127–143). Amsterdam: John Benjamins Publishing.

Webster, M. (2007). Merriam Webster's rhyming dictionary. Merriam-Webster, Inc. Springfield, Massachusetts.

Zhang, Z., Gentile, A., & Ciravegna, F. (2012). Recent advances in methods of lexical semantic relatedness – A survey. *Journal of Natural Language Engineering, 19*(4), 411–479.

Zock, M. (2015a). 'Errare humanum est'. Refusing to 'appreciate' this fact could be a big *mistake*! In G. Adda, M. Adda-Decker, J. Mariani, V. Barbu Mititelu, D. Tufis, & I. Vasilescu (Eds.), *Errors by Humans and Machines in multimedia, multimodal and multilingual data processing. Proceedings of ERRARE 2015*. Bucharest: Romanian Academy Publishing House.

Zock, M. (2015b). *Introduction* to the special issue of 'cognitive aspects of natural language processing' (Words in books, computers and the human mind). *Journal of Cognitive Science, 16*(4), 355–378. Institute for Cognitive Science, Seoul National University (http://j-cs.org/gnuboard/bbs/board.php?bo_table=__vol016i4).

Zock, M., & Biemann, C. (2016). Towards a resource based on users' knowledge to overcome the Tip-of-the-Tongue problem. In *Proceedings of the COLING Workshop 'Cognitive Aspects of the Lexicon' (CogALex-V)* (pp. 57–68) Osaka, Japan.

Zock, M., Ferret, O., & Schwab, D. (2010). Deliberate word access: An intuition, a roadmap and some preliminary empirical results. *International Journal of Speech Technology, 13*(4), 107–117.

Zock, M., & Schwab, D. (2011). Storage does not guarantee access. The problem of organizing and accessing words in a speaker's lexicon. *Journal of Cognitive Science, 12*(3), 233–258. Institute for Cognitive Science, Seoul National University.

Zock, M., & Tesfaye, D. (2015). Automatic creation of a semantic network encoding *part_of* relations. *Journal of Cognitive Science, 16*(4), 431–491. Institute for Cognitive Science, Seoul National University.

Zock, M., Wandmacher, T., & Ovchinnikova, E. (2009). Are vector-based approaches a feasible solution to the « tip-of-the-tongue » problem? S. Granger & M. Paquot (Eds.), *eLexicography in the 21st century: New challenges, new applications* (pp. 355–366). Louvain-la-Neuve.