WordNet and beyond : the case of lexical access

Michael Zock
AMU, LIF, UMR 7279
163, Avenue de Luminy
13288 Marseille / France
zock@free.fr

Didier Schwab
Univ. Grenoble Alpes
LIG - GETALP
Campus de Grenoble / France
didier.schwab@imag.fr

Abstract

For humans the main functions of a dictionary is to store information concerning words and to reveal it when needed. While *readers* are interested in the meaning of words, *writers* look for answers concerning usage, spelling, grammar or word forms (lemma). We will focus here on this latter task : help authors to find the word they are looking for, word they may know but whose form is eluding them. Put differently, we try to build a resource helping authors to overcome the tip-of-the-tongue problem (ToT).

Obviously, in order to access a word, it must be stored somewhere (brain, resource). Yet this is by no means sufficient. We will illustrate this here by comparing WordNet (WN) to an equivalent lexical resource bootstrapped from Wikipedia (WiPi). Both may contain a given word, but ease and success of access may be different depending on other factors like quality of the query, proximity, type of connections, etc. Next we will show under what conditions WN is suitable for word access, and finally we will present a roadmap showing the obstacles to be overcome to build a resource allowing the text producer to find the word s/he is looking for.

1 Introduction

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 him to look deliberately and often painstakingly for a specific, possibly known word (Zock et al. 2010; Abrams et al. 2007; Schwartz, 2002; Brown, 1991).

We will be concerned here with this latter situation. More specifically, we are 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 dictionaries 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 [1], and only the *end point* (outputs) the corresponding target word. While it is true that most dictionaries have been built with the reader in mind, one must admit though that attemps have been made to assist also the writer. The best known example is probably Roget's Thesaurus (Roget, 1852), but as we will see, there is also WordNet (Miller, 1990; Fellbaum, 1998) [2], a very special kind of resource integrating in a single place information 'normally' spread over different dictionaries. Rather than creating different volumes for different tasks (allowing the user to find a definition, synonyms, antonyms, etc.), WordNet (WN) has integrated all these functions into a a single resource. As its spirit is closest to what we have in mind, we will focus

[1] 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).

[2] For other pointers to onomasiological dictionaries, see (Zock et al. 2010).

on it in this paper, commenting on its strenghts and weaknesses with respect to word access.

This paper is organized as follows. We 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* (henceforth, ToT) [3]. We 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 our claim we will compare an extended version of WN (Mihalcea and Moldovan, 2001) to an equivalent resource based on Wikipedia.

Next, we will discuss under what conditions WN is adequate for word access, and finally, we will sketch a roadmap describing the steps to be performed in order to go beyond the Princeton resource. The goal is to build an index (association network) and navigational tools (categorial tree) to help authors to find the word they are looking for when being in the ToT state.

2 Storage does not guarantee access

To test this claim we ran a small experiment, comparing an extended version of WN 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 guaranteee 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) [4]. 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 [5], 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).

2.1 Examples and comparisons of the two resources

The table here below shows the results produced by *eXtended* WN and WiPi for the following, randomly given inputs : 'wine', 'harvest' or their combination 'wine + harvest'.

Input:	*Output* : eXtended **WN**	*Output* : **WiPi**
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 (3d), ...

Table 1: Comparing two corpora with various inputs

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 though is the fact that none of these terms allows us to access the target, eventhough it is contained in the database of *x*WN, which clearly supports our claim that storage does not guarantee access. Things are quite

[3] 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.

[4] To show the relative efficiency of a query, we have developed 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. 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.

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

different for an index built on the basis of information contained in WiPi. 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.

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 an index allowing users to find the elusive word.

One may wonder why we failed to access information contained in WN and why WiPi 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 WiPi), and the number of syntagmatic links, both of which are fairly small compared to WiPi. Obviously, being an encyclopedia, WiPi 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). True as it is, this would nevertheless not affect our line of reasoning or our conclusion. Even in a larger lexical resource we may fail to find what we are looking for because of the lack of *syntagmatic links*. 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 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).

Another noteworthy point is the fact that success may vary quite dramatically, depending on the input (quality of the query). As Table 2 shows, WN outperforms WiPi for the words 'ball', 'racket' and 'tennis'. Yet, WiPi 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).

Input:	*Output* : eXtended **WN**	*Output* : **WiPi**
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 (3d), ...

Table 2: Comparing two corpora with various inputs

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 WiPi one may well consider integrating the two by relying on a resource like *BabelNet* (Navigli and Ponzetto, 2012) [6]. 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.

3 Under what condition is WN really good for consultation ?

Many people know that WN is based on psycholinguistic principles. What is less known though is the fact, that despite its psycholinguistic 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 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 certains circumstances.

Remains the question under what conditions WN is able to reveal the elusive target word. We 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]+*synonym* = [?] → [bitch]);
([dog]+*hypernym* = [?] → [canine]);

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

(b) the *input* (source word) and the *target* are *direct neighbors* in the resource. For example,
[seat]-[leg] (*meronym*);
[talk]-[whisper] (*troponym*), ...

(c) the *link* is *part* of WN's database, e.g.
'hyponym/hypernym', 'meronym', ...

4 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 the word and, if he cannot find it, he may nevertheless be aware of some other word(s) somehow connected to the target. This is a very important aspect, as we will start from that.

When we wrote that WN is quite successful with regard to word look-up under certain circumstances, we also meant to say 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, it will definitely reveal the target [7].

(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 same part of speech, word access is difficult if the output (target) belongs to a different part of speech than the input (prime) [8];

(d) the user ignores the link, he cannot name it, or the link is not part of WN's repertory [9]. Actually this holds true (at least) for nearly all syntagmatic associations;

Let us see how to go beyond this. To this end we 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 us clarify some differences between hierarchically structured dictionaries and our approach.

While lexical ontologists (LO) try to integrate all words of a language into a neat subsumption hierarchy, we try to group them only in terms of direct neighborhood, not mentioning at all the type of the link. Words are grouped later on by category (see, figure 1). This yields a quite different network than WN. Our graph is fully connected and, not being concerned with exhaustivity, we try to reveal only the words typically evoked by some input. This being so, our graph (or, any equivalent association network) will yield different results than WN for the same input (see table 3).

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

Table 3: The respective outputs produced by a *lexical ontology* (here WN) as opposed to an *association network* (here, the E.A.T).

Suppose we started from a broad term like 'food'. A LO 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 also syntag-

[7] 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.

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

[9] For example : 'well-known_for', 'winner_of', ...

matically related words (*origine* : France; *state* : hungry, ...). Compare the results obtained by WN and the Edinburgh Association Thesaurus [10].

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 '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. While the *semantic map* defines the territory within which search takes place, the *lexical compass* guides the user, helping her or him 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 we 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 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 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 either knows the word (in which case the problem is solved), or he does not. In this latter case all he can do is to rely on available knowledge concerning the target, an assumption we make here. Knowledge is fragmentary. Yet, incomplete as it may be, this kind of information may allow us to lead him to 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 hours. 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 [11]. 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

[10] http://www.eat.rl.ac.uk

[11] 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]}).

goal, i.e. target word [12], 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.

Let us now see quickly how to make this idea 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), we suggest a dialog between the user and the computer to reduce incrementally the search space. The user provides the input [13], — word coming to his/her mind, generally a word more or less directly related to the target,— and the system makes a set of proposals (list of words), trying to guide the user on the basis of her input.

Suppose that the target were 'gull'. In such a case one might ask : 'do you know the name of a bird able to swim', having yellow feet, and a long beak [14]? To simplify matters and to convey as simply as possible the rationale underlying our approach (see figure 1, next page), let us assume that the input is a single word. The process consists basically in the following steps : (a) user input (query), (b) system output (answer), (c) user's choices concerning the target (does the list contain it?), or, choice of the word to continue search with. Concretely speaking this leads to the following kind of dialogue. The user starts by providing her input, that is, any word coming to her mind, word somehow connected to the target (step-1, figure 1) [15]. The system presents then in a clustered and labeled form (categorial tree) all direct associates (step-2, figure 1) [16]. 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 provides an entirely new word and the whole process iterates. The system will come up with a new set of proposals.

As one can see, this 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

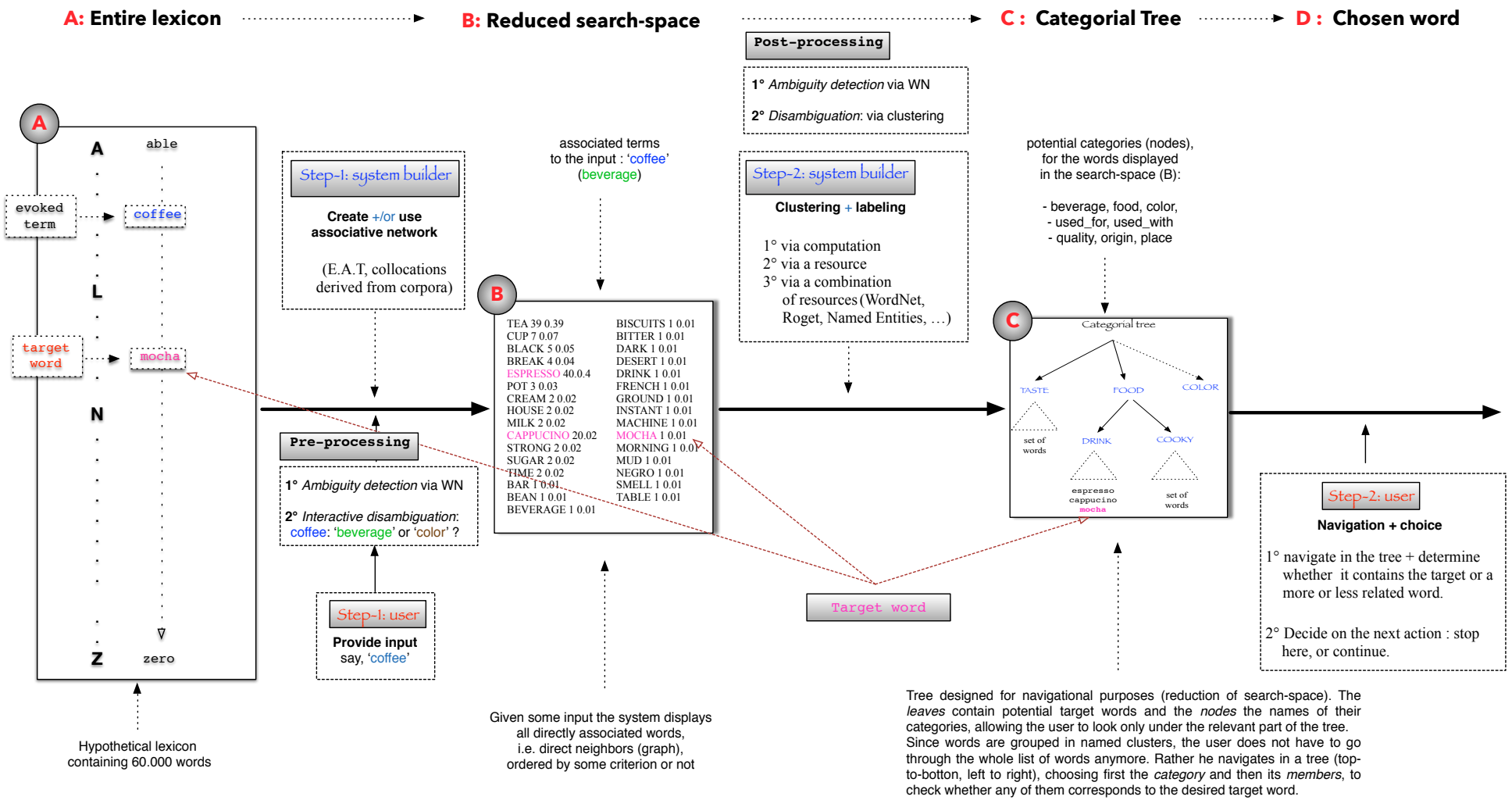

Figure 1 : Lexical access as a two-step dialogue

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 sum, categories act somehow like signposts signaling the user the kind of words he is likely to find going one way or 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 [17], 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 are really useful) [18]. To put words into clusters is one thing, to give them names an ordinary dictionary user can understand is quite another [19]. 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, we think 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 topic or to strongly related problems (Zhang et al., 2012; Biemann, 2012; Everitt et al., 2011).

5 Conclusion

We have started the paper by pointing out the fact that word access is still a problem for dictionary builders and users (see also Thumb, 2004), in particular humans being in the production mode (Zock, 2015). Next, we showed that the fact that an item is stored in a lexical resource does not guarantee its access. We continued then to discuss why even a psycholinguistically motivated resource like WN often fails to reveal the word authors are looking for.

Finally, we presented a roadmap to overcome this problem. The idea is to build a resource guiding a human user allowing him to find the word he is looking. Given some input (user's knowledge concerning the target word), the system would provide the direct neighbors in a clustered and labeled form (output) to allow the user to check whether this tree contains the elusive word. While the system's task with respect to the user's input (step-1) is to reduce search space, the function of the second step 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 we 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.

[17] 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’.

[18] Take for example the Wikipedia page devoted to 'Panda', and check which of the co-occurrences are those typically evoked when looking for the word 'Panda'.

[19] 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.

Reference

Abrams, L., Trunk, D. L., and 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.

Aitchison, J. (2003). Words in the Mind: an Introduction to the Mental Lexicon. Oxford, Blackwell.

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

Bilac, S., Watanabe, W., Hashimoto, T., Tokunaga, T. and Tanaka, H. (2004). *Dictionary search based on the target word description*. In: Proc. of the Tenth Annual Meeting of The Association for Natural Language Processing (NLP2004), pages 556-559.

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

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

Dutoit, D. and P. Nugues (2002): *A lexical network and an algorithm to find words from definitions*. In Frank van Harmelen (ed.): ECAI2002, Proceedings of the 15th European Conference on Artificial Intelligence, Lyon, pp.450-454, IOS Press, Amsterdam.

El-Kahlout I. D. and K. Oflazer. (2004). *Use of Wordnet for Retrieving Words from Their Meanings*. 2nd Global WordNet Conference, Brno

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

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

Mihalcea, R. and D. Moldovan, (2001): Extended WordNet: progress report. In NAACL 2001 - Workshop on WordNet and Other Lexical Resources, Pittsburgh, USA.

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

Miller, G.A. (ed.) (1990): WordNet: An On-Line Lexical Database. International Journal of Lexicography, 3(4), 235-244.

Navigli, R. and Ponzetto, S. (2012), BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. Artificial Intelligence, 193, pp. 217-250

Rapp, B. and Goldrick, M. (2006). Speaking words: Contributions of cognitive neuropsychological research. Cogntive Neuropsychology, 23 (1), 39-73

Roget, P. (1852). Thesaurus of English Words and Phrases. Longman, London.

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.

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

Thumb, J. (2004). Dictionary Look-up Strategies and the Bilingualised Learner's Dictionary. A Think-aloud Study. Tübingen: Max Niemeyer Verlag.

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

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

Zock, M., Ferret, O. and Schwab, D. (2010) Deliberate word access : an intuition, a roadmap and some preliminary empirical results, In A. Neustein (Ed.) *'International Journal of Speech Technology'*, Springer Verlag, 13(4):107-117.

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