Automatic index creation to support navigation in lexical graphs encoding *part_of* relations

Michael Zock[1] *Debela Tesfaye*[2]
(1) LIF-CNRS, 163 Avenue de Luminy, 13288 Marseille, France
(2) ITPHD PROGRAM, Addis Ababa University, Addis Ababa, Ethiopia
michael.zock@lif.univ-mrs.fr, dabookoo@yahoo.com

ABSTRACT

We describe here the principles underlying the automatic creation of a semantic map to support navigation in a lexicon, our target group being authors (speakers, writers) rather than readers. While machines can generally access information that it has stored, this does not always hold for people. A speaker may very well know a word, yet still be (occasionally) unable to access it.

To help authors to overcome word-finding problems one could add to an existing electronic resource an index based on the (age-old) notion of association. Since ideas or their expressive forms (words) are related, they may evoke each other (lemon-yellow), but the likelihood for doing so varies over time and with the context. For example, the word 'piano' may prime 'instrument' or 'weight', but which of the two gets evoked depends on the context: 'concert' vs. 'house moving'. Given this dynamic aspect of the human brain, we should build the index automatically, computing the relation of terms and their weights on the fly. This dynamic creation of the index could be done via a corpus. This latter representing ideally the dictionary users' world knowledge, and the way how the prominence of words and ideas varies over time.

Another important point are link-names, i.e. the type of relationship holding between two associates: [(rose) <--color (red)]. Given the fact that any query (e.g. 'India') may yield many hits, hits whose weights may be misleading, it makes sense to group the output according to some (other) category, for example, link names (color, city_of, instrument, ...). Yet, important as they may be, links or relations are hard to extract and to name. This is why we have decided to start with a very small sub-set, meronymic-, i.e. part-of relations (x is *part of* y, x *has* y, etc.).

KEYWORDS : Lexical access, navigation, word association, lexical graphs, semantic maps, automatic index creation, dynamic index, link extraction, link-names, part-of relations.

1 Introduction

One of the most vexing problems in speaking or writing is the fact that one has memorized, i.e. stored a given word, yet one fails to access it when needed. This kind of search failure known as *dysnomia* or *Tip of the Tongue-problem* (TOT),[1] occurs not only in language, but also in other activities of everyday life. It is basically a search- and index problem which we are reminded of when we look for something that exists in real world or our mind (keys, glasses, people's names), but which we are unable to locate, access or retrieve in time.

[1] The TOT-problem is characterized by the fact that the author (speaker/writer) has only partial access to the word s/he is looking for. The typically lacking parts are phonological (syllables, phonemes). Since all information except this last one seems to be available, and since this is the one preceding articulation, we say: the word is stuck on the *tip of the tongue*.

Proceedings of the 3rd Workshop on Cognitive Aspects of the Lexicon (CogALex-III), pages 33–52,
COLING 2012, Mumbai, December 2012.

Word finding problems are generally dealt with via a lexicon. Obviously, readers and writers have different behaviors and expectations concerning input and output (target information). While the *decoder* (listener/reader) provides the word s/he wants additional information for —(say, what is the meaning of 'rug', or what are its synonymes?),— the *encoder* (speaker/writer) provides the meaning, or meaning-related elements (for example, 'typical british sport') of the word for which s/he lacks the corresponding form (=> cricket).

Our concern here is more with the language producer, *i.e.* lexical access in language production, a task often neglected in lexicographical work. Language producers typically start from meanings (concepts) or lexical items related to the target word: associations (strong + black + bitter + beverage + made_from beans => coffee). Eventhough empirically well founded, *concept-based search* or access via associations (Deese, 1965; Schvaneveldt, 1989) is not well supported in current electronic dictionaries. Actually, there are several problems to be addressed, let us mention only two: (a) the problem of *input*: how (i.e. in what terms) shall the user specify the meaning of the word whose form he is looking for? —(say, 'name of the beverage the British fancy to take in the afternoon'),— and (b) the problem of *navigation*. How do you get from some input (*source word*), —say, 'huge animal, gray, trunk, ivory, Africa',— to the *target word* (elephant)? Note that studies concerning the TOT-problem have shown over and over that people being in this state know a lot concerning the target word —meaning, origin, gender, number of syllables, etc.,— even if they cannot access its form (Brown,1991; Brown and Mc Neill, 1966).

2 Creation of an association-based index

To support word finding, i.e. navigation/word access in electronic dictionaries, Zock and colleagues proposed to add to an existing electronic resource a corpus-derived index based on the notion of association (Zock et al., 2010). Dictionary entries (headwords), say 'rose' or 'book', are indexed in terms of the words they evoke: rose => red or flower; book => bible or library, ... This kind of information can be gleaned via various methods, including corpus analysis, i.e. collocation-extraction (Ferret and Zock, 2006). Words co-occurring in a given text —the window being generally a sentence or a paragraph at the most— can be considered as associates. They tend to evoke each other. Note that associations can be bi-directional, though their strength and link-type are hardly ever the same. The list of co-occurrences can be represented in various ways, lists, graphs, etc. They can be seen as a special kind of semantic network (Sowa, 1992). Indeed, the links are hardly ever deep-case roles (agent, beneficiary, etc.), but rather associations, i.e. binary relations.

The fact that the index is a network has various interesting features. It provides agents (people, robots) with a powerful search tool, while offering a lot of freedom, i.e. flexibility. Since all words are connected, any of them can be the source (prime, potential starting point) or the target (probe). Search can start at any point, i.e. all words can be reached from anywhere, regardless of their distance (indirect neighborhood). Even if search has been initiated from a remotely related word, one may still be able to find the target word. One just has to use (recursively) one of the query's associates (direct neighbour) as new starting point. Since all words can act as retrieval cue, all of them trigger at least one related word, and if they trigger more, that is, a list of words (they usually do), it may contain the target word, and if not, a word indirectly related to it.

The idea of association is of course not new. It was known already to the Greek philosophers and it has a quite a long tradition in psychology (Aitchison, 2003; Galton, 1880) More recently it has

been used in computational linguistics (vector-based approaches: Landauer and Dumais, 1997 ; Lund and Burgess, 1996) and computational lexicography, lexical graphs like WordNet (Miller, 1990). It should be noted though that many lexical graphs lack a vital piece of information, the link type (synonyms, hypernym, etc.). Yet this is vital information, as we will see (section 3.3) at the interface level for human users. Concerning WordNet (WN)[2], it should be pointed out that links are all hand-coded (see section 5), and the resource is not corpus-based, hence it lacks many of the needed links, mostly syntagmatic associations. WN suffers from the well-known 'tennis-problem': words typically occurring together, hence naturally associated (tennis, umpire, racket, court, backhand), are not always linked in WN. Before discussing this last point, the core of the paper, let us describe briefly the method used for building the index and some of the problems.

3 Building the resource

Creating a dictionary involves typically the following decisions: (a) which words to include (this raises the problem of what a word is); (b) what *information* to *associate* with each one of them (definition, grammatical information, ...); (c) how to organize the lexicon, i.e. lexical entries (alphabetically, topically). Of course, all these decisions depend to a large extent on the subsequent usage of the resource (reading, writing).

The resource we have started to build is a kind of semantic map, with words being connected, and the links (or connections) being typed (categorized, labeled) and weighted. Of course, there are various methods to build such a map. One way is to ask people to get lists of associations (Deese, 1965). This has been the main strategy of psychologists trying to define word association norms (Nelson et.al., 1998). Another way is to use games (Lafourcade, 2007). Still another approach is to use corpora and to extract collocations. This is the route we are taking. Yet, in order to teach our goal several problems need to be addressed:

3.1 Building a representative corpus:

Since we start from the assumption that peoples' associations are based on specific- and on general knowlege (episodic- and encyclopedic knowledge), we must make sure that this kind of information is also contained in the sources upon which we draw in order to build our lexical map (association lists). Put differently, our sources (in our case corpora) must be representative. To this end we need a well balanced corpus, that is a corpus containing general information (for example, London, capital of England, etc) as well as information concerning a specific person, place or event.

3.2 Indexing:

In order to find the words a dictionary contains, we must organize them. Put differently, the resource must be structured, i.e. it must contain an index or a semantic map. Words can be organized according various criteria or viewpoints, *formal*-syntactic (spelling, part-of-speech, morphemes), pragmatic-*semantic*, etc. In this latter case one may consider (a) the word's components, i.e. the elements occuring in a word's definition (bag of word: Dutoit et al. 2002; Bilac et al. 2004; El-Khalout et al. 2004), (b) its recurring elements (semantic primitives (Schank,

[2] http://wordnet.princeton.edu

1975; Wierbicka, 1996) or (c) its role in discourse: words are grouped by domain, (see Roget's Thesaurus, Roget, 1852).

Unlike linguists, psychologists are more interested in word relations.Gathering typically related terms (x evoking y) they've built association lists (Deese, 1965; Schvaneveldt, 1989). Such lists are nowadays freely available in different languages : Dutch,[3] English ([4],[5]), French[6], German,[7] Japanese,[8] and Russian.[9] The Edinburgh Associative Thesaurus is particularly interesting, in as it shows not only the words evoked ('red', 'flower', etc.) in response to a given stimulus ('rose'), but also the causes (primes) of this input. For example, 'thorn', 'petal', 'flower', etc. in response to the prime 'rose'. Put differently, we get bi-directional, i.e., incoming and outgoing links. While such resources are extremely useful for many tasks (pratical applications, research), they nevertheless do have certain shortcomings. For example, they are fairly static. Hence, they cannot take topic changes into account. Yet, associations are sensitive to such variations. Think of the word 'piano' in the context of moving from one place to another. Also, most of these resources lack the link type, yet this is an important feature to reduce search time by clustering information pertaining to the same link type. This last comment does not apply to WN or JeuxdeMots. They both contain a small set of link types[10] which is very useful for navigation.[11]

3.3 Ranking:

Words occur with a certain frequency. The same holds true for their combination, that is, words and their relations do have a certain weight. While one should not overestimate the notion of weight with respect to word access, it may nevertheless be useful for word order (priorization of words in the list of candidates) and for deciding where to draw the line (cut-off point in case that the list gets long), that is which words to display and which to hide. Ideally, the weight is (re-) computed on the fly, taking into account contextual variations. As mentionned already in our piano example, a word may give prominence to quite different associations depending on the context. Likewise, the word 'Java' may evoke in people's mind quite different concepts ('island' or 'programming language') depending on whether we are talking about holidays, geography or computers.

3.4 Identification and 'typing' the links:

Associations must not only be identified, they must also be labeled. Qualifying, i.e. typing the links is the hardest task, yet it is vital for navigation. Frequency alone is not only of limited use

---

[3] http://www. kuleuven.be/semlab/interface/index.php

[4] Edinburgh Associative Thesaurus : http://www.eat.rl.ac.uk/

[5] University of South Florida Word Association: http://w3.usf.edu/FreeAssociation/

[6] JeuxdeMots: (www.lirmm.fr/~lafourcade/jeuxdemots/diko.php)

[7] http://www.coli.uni-saarland.de/projects/nag/, http://www.schulteimwalde.de/resources/assoc-norms.html

[8] http://www.valdes.titech.ac.jp/~terry/jwad.html

[9] http://wordassociations.ru

[10] JeuxdeMots contains the following links (isa, hyponyme, syonyme, antonyme, domain, substance, location, caracteristics, part_of, meronym, quantifier, do, cause, consequence) plus a 'link' called: 'free association'.

[11] AKI: http://www.jeuxdemots.org//AKI.php

(people cannot interpret properly numerical values in a context like this),— it can even be misleading. Two terms of very similar weight, say, 'mouse' and 'PC', may belong to entirely different categories: 'computer device' vs. 'type of computer'. Hence choosing one instead of the other may decrease the chances of finding the desired target-word. In the same vein, BLACK(x) may be strongly associated with WHITE(x), DARK(x) and COFFEE(x), eventhough its relationship may be quite different in each case: 'opposite', 'similar to' and 'color'. Last, but not least, 'right' may be strongly associated with 'write', 'light', 'left' or 'wrong' which, of course, does not imply that the relationship is the same.

Note, that weights are a main feature in the programs written by psychologists where they try to mimick the performance of the human brain, or, the mental lexicon. The goal is to mimick *precision* (correct output, or similar errors to the ones produced by people) and *access time* (word access in real-time). This work is generally done within the connectionist framework (Dell, 1996; Levelt et al., 1999). Impressive as these simulations are, this approach cannot be used here for several reasons: (a) the information encoded in these networks is not interpretable by human user. Actually, the information contributing to the 'building' of a word,—words are synthesized rather than stored,— is distributed across various layers[12]; (b) the weights are tuned by the system builders (psychologists) who know the final output (target word). This does not hold for the user of our future resource, since target word is precisely the item s/he is looking for, and if s/he knew it, the problem were solved.

4 Some justifications for making explicit the link-type

As mentioned already, a lexical graph composed of words only is of little use for navigation, if one does not know the kind of link holding between two adjacent nodes (direct neighbors, i.e. associated words). Indeed, every node, i.e. every word may have a great many associates, some being linked via the same type of association —(imagine all the days subsumed under the label 'weekdays' or 'colors'),— others being connected via a link of a different type (*week*-month; *week*-weak, *week*-geek, etc.).

Obviously, the greater the number of words associated with a term, and the more numerous the type of links, the more complex the graph will be. This reduces considerably the value of graphs as adequate representation at the interface level in order to support navigation. There are at least three potential problems challenging readability:

- **High connectivity** (i.e. the great number of possible links). These links can be of different types, bi-directional (incoming and outgoing), asymmetric and of different weights.
- **Distribution** (i.e. non-adjacency, of conceptually related nodes, that is, nodes activated by the same kind of association (e.g. synonyms), but not being displayed next to each other.
- **The possible crossing of links** in the case of indirect association (see A1 – B2 or A2 – B1 in Figure 1, next page).[13]

[12] For a detailed description, see Zock et al., 2010.

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

All these factors may leed to confusion. Also, the role of frequencies must be relativized or defined more precisely. Indeed, many researchers believe that frequencies or weights are the crucial element for guiding search. Yet, taken alone they are too poor to guide the user, helping him to decide on the direction to go (see section 3.4).

In sum, lexical graphs can become complex, not only because of the number of nodes (words they contain), but also because of the number of possible connection types (associates). Hence, lexical graphs devoid of this kind of information are like maps that omit showing 'how' cities are connected (road, railway, airplane). Hence, they are not sufficiently good representations of the territory (semantic map) to be used as orientational guides or navigational aids.

To overcome these problems, we suggest to display by category (clusters) all the words linked by the same kind of association to the source word. Hence, rather than displaying all the connected words as a flat list, we suggest to present them in chunks to allow for categorical search. Having chosen a category, the user will be presented a list of words or categories from which he must choose. If the target word is in the category chosen by the user —(suppose he looked for a hyperonym, hence he checked the is_a bag),— search stops, otherwise it goes on. The user could choose either another category, or a word in the current list, either of which becoming then the new starting point.

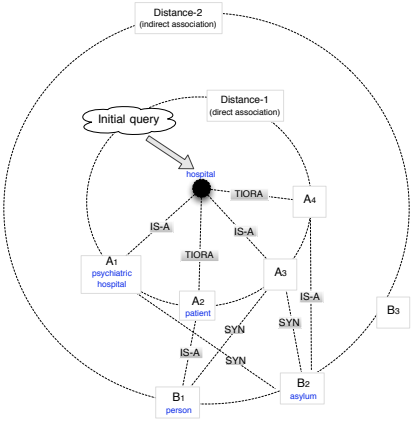

FIGURE 1-Potential problems with graphs: crossing links with indirect neighbors.[14]

In the next section we will present some initial results of how to infer automatically the type of link for a small subset of links: part_of relations.

[14] IS-A (subtype); SYN (synonym); TIORA ('Typically Involved Object, Relation or Actor', for example, tools, employees, ...).

5 Initial results for inferring automatically the type of link

Suppose that you wanted to express the following concept: '*superior dark coffee made of beans from Arabia'*, and that neither 'espresso' nor 'cappuccino' are the desired target words. In this case there are three kinds of relations likely to help the language producer find the target word 'mocha'. Indeed, the mentioned seed words (superior, dark, coffee, made of, beans, from Arabia) express different kind of relations: an *attribute* relation (superior, coffee; dark, coffee), a *resulting* relationship (coffee made of beans) and a *source* relation (from Arabia). Aggregating them and using them as retrieval cues might help the language producer to narrow down the search space, zooming into a small set of words possibly containing the target word. To allow for this, we need, of course, something like a semantic map. This latter specifies the form of the major words and the way how they are related to their direct neighbors. Such a map can reveal many things: list of available words, distance between two words, type of relations, relative density, i.e. tightly populated parts of the network, hubs, i.e. number of incoming and outcoming links, etc.

Starting from such a set of seed- or source words, Zock and colleagues (Zock et al., 2009) have used LSA and the Tf-idf measure values to identify the target word. LSA is quite successful with respect to identifying the relative similarity between concepts. Actually, it achieves similar scores as non-natives do: 64% vs. 64,5% (Landauer and Dumais, 1997). While this is surely impressive, LSA cannot provide us with the kind of information we care for: the name of the relationship holding between two concepts or words. Actually, our problem is a bit different from the one addressed by LSA. Our goal does not consist in finding synonyms of the source- or target-word, our goal is to help people to *find* the *target word*, bottom-line. In other words, we need a different approach. For example, our system should be able to draw on any information available at the onset of search. Hence, search should be possible by entering the graph at any point. Also, our associations must not only be identified as in LSA or lexical graphs in general, they must also be labeled in terms of their type. As mentioned already, this is a prerequisite if we want to help humans to navigate in the semantic space for which we try to build a map.

To achieve this goal we will draw on the idea described in section 2. The problem of developing such a semantic space is enormous as there are many kinds of relations needed, for example: Cause-Effect (laugh-wrinkles), Product-Producer (honey-bee), Content-Container (wine-bottle), Part-Whole (tip-tongue), Instrument-Agency (laser-printer), etc. We will focus here only on one of them, Part-Whole relations (PT-WHRs) and their automatic extraction from corpora to build the semantic map or space. Several scholars have proposed taxonomies of PT-WHRs (Winston et al., 1987; Vieu and Aurnague, 2007). We will follow Winston's classical proposal:

1.	component – integral object	handle – cup
2.	member – collection	tree – forest
3.	portion – mass	grain – salt
4.	stuff – object	steel – bike
5.	feature – activity	paying – shopping
6.	place – area	oasis – desert

- *Integral objects* have a structure; their components can be torn apart, and their elements have a functional relation with respect to the whole. For example, 'kitchen–apartment' or 'aria–opera.
- 'Tree-forest' and 'chairman-committee' are typical representatives of Member-Collection relations.

- Portion-Mass captures the relations between portions, masses, objects and physical dimensions. For example: 'meter-kilometer'.
- The Stuff-Object category encodes the relations between an object and the stuff of which it is made of. For example, 'steel-car' or 'snow-snowball'.
- Place-Area captures the relation between an area and a sub-area like 'Ethiopia-Addis Ababa'.

Meronymic relations can also be categorized as typical or accidental. The former are always true (roof-house), while the latter are episodic (cucumber-sandwich), they have happened only at some point in time. We focus here only on the first type.

To capture the meaning of words we relied on the intuition that meanings depend to some extent on a word's neighbourhood, be it direct (black coffee) or indirect (the color of coffee is generally black). Words occurring in similar contexts tend to have similar meanings (Harshman, 1970). This idea, known as the 'distributional hypothesis,'[15] has been proposed by various scholars (Harris, 1954; Firth, 1957; Wittgenstein, 1922). It implies that word meanings are context sensitive. A word's meaning cannot be fully grasped unless one takes the context into account. Meaning and context can be captured in terms of (more or less direct) neighbourhood, i.e. words co-occurring within a defined window (phrase, sentence, paragraph).

5.1 Description of our approach

Since we try to capture meaning via word similarity, the question arises of how to operationalize this notion. One way of doing so is to create a vector space composed of the target word and its neighbours (Lund and Burgess, 1996). This approach, known as vector space model (VSM) has been developed by Salton and colleagues (Salton et al., 1975) for information retrieval. Their idea was to represent all documents of a collection as points in a space, i.e. a vector in a vector space. Semantic similarity is expressed via the distance of two points: closely related points express similarity, while distant points signal unrelated ideas, or remotely related words. We are concerned here with word similarity rather than document similarity. The meaning of a word is represented as a vector based on the n-gram value of all co-occurring words. The use of the VSM to extract PT-WHRs has two advantages: it requires little man power (human effort) and few resources (corpora), at least far less than Girju's approach (Girju et al., 2005) which relies heavily on annotated corpora and WN.

The underlying idea is that the type of relation holding between two concepts/words can be inferred from data (for example, corpora containing co-occurrences), by using the similarity values and n-gram information for clustering the relevant terms. The similarity value allows us to extract part_of relations, while clustering is used to group similar words. The similarity value can be obtained in different ways, and it may depend on the type of relation to be identified. Put differently, the vectors used for encoding, say, a part-whole relation are different from those encoding hyponyms. The n-gram information used to extract the vectors is also specific to the type of semantic relation to be encoded.

[15] http://en.wikipedia.org/wiki/Distributional_hypothesis

We devised a weakly supervised method for automatic extraction of meronymic relations (component–integral object; part-whole).[16] Indeed, our method hardly depends on language and it is completely domain-independent. However, we do need a 'Part of Speech Tagger' or a 'part-of-speech tagged corpus'. In this respect our work differs quite a bit from other people's work as it does not require a resource like WN. Hence, our approach can be used even for under-resourced languages, or languages lacking a resource like WN. In other words, the methods is sufficiently general to be applicable to other languages than the one for which it has been initially designed.

Since word-meaning is represented as a vector based on the n-gram value of all co-occurring words we need a corpus. To build the required vectors we used the 'Corpus of Historial American English' (COHA) which contains 400 million words. COHA is an n-gram corpus tagged for parts of speech (Mark, 2011). For languages lacking this kind of (tagged) corpus, plain text can be used, as the system is able to identify the concepts' n-gram value in the corpus. This feature is very convenient for under-resourced languages, as it makes their preparation (pre-processing) easier than if one had to annotate the corpus manually.

5.2 Related works

Previous works attempting to identify semantic information are somewhere on a scale, ranging from exclusively hand-crafted patterns (Hearst, 1998) and rules to probabilistic methods. For example, Finin (1980) relied exclusively on manually built rules. Girju (2005) and Beamer (2008) used a knowledge intensive approaches by drawing on huge resources like WordNet. Hage's (2006) and Harshman's work (1970) is domain dependent, while the proposals of (Girju, 2005; Matthew & Charniak, 1999) rely on syntactic structures, hence they are language specific.

Resource intensive approaches (like the ones relying on WordNet) are not suitable for languages lacking such a resource, for example, under resourced languages. Resource intensive approaches use texts, tagged with WordNet information, for example, senses. However, this kind of approach cannot be applied to applications relying on real world data, real world texts are hardly ever tagged with WordNet information (senses, type of link, etc.). In addition, most of the above mentioned approaches are highly language dependent. The classification features used to build the rules are extracted from a specific language. For example, Hearst (1998) uses syntactic features that occur frequently in sentences and in many kinds of texts. However, such syntactic structures are rare, their coverage is small and their effectiveness greatly depends on the type of semantic relation extracted. Indeed, Hearst (1998) reported better results for *hyponyms* than for *meronyms* which may be due to the fact that syntactic structures encoding this latter kind of relation tend to be ambiguous. This being so we may need to take a different approach.

We decided to use a Vector Space Model (VSM) which was highly successful for various applications, including question-answering. For instance, using this kind of approach for representing word meaning Rapp [38] achieved a score of 92.5% on multiple-choice synonym questions from the TOEFL Test (the test foreigners have to take to evaluation their level of English before entering an american university[17]), whereas the average human score for non-

[16] *Supervised* learning means that the examples on the basis of which the system learns are *labeled*, i.e. they specify explicitly which forms are correct and which are not. In *unsupervised* learning examples are not *labeled*, the system clusters data into classes, giving the latter some arbitrary name.

[17] http://www.ets.org/toefl

native speakers was 64.5%. Motivated by this success we decided to try this approach for automatically extracting part_of relations.

5.3 Our approach in more detail

As explained in section 3, four problems need to be solved for building the resource. We need to get a representative corpus, index lexical entries in terms of associations (i.e. build an association matrix), rank the terms and label the links.

To address the first task we used the Brown corpus, though, other corpora are probably needed. Next, we developed a system, i.e. a pipeline of 6 stages or components (see figure 2) to address the remaining problems. The process works as follows. Starting with the first word in the corpus, the system extracts all associated words expressing a PT-WHR to continue then with the next word until it has reached the end. Actually, the system performs the following six operations:

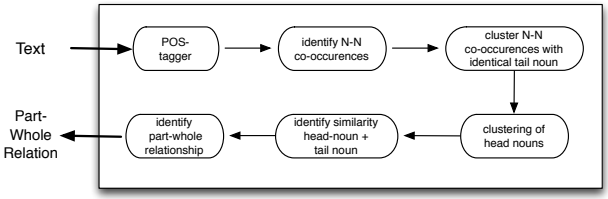

FIGURE 2- System information flow

- *Step 1*: This component identifies the part of speech of the sentence elements. Since *part-whole relations* connect only nouns, the system requires only a tagger able to identify nouns. As mentioned already, we used the 'Corpus of Historical American English' (COHA) (Mark, 2011). This is an n-gram corpus whose elements are tagged in terms of part of speech.
- *Step 2*: The next component extracts Noun-Noun co-occurrences (N-N sequences) from the tagged corpus. For example, ‘corolla car’, ‘door of car’, ‘car engine’, ‘engine of car’, ‘car design’, ‘network design’, ’airplane engine’, ‘search engine’ etc. Noun phrases are not included in our current version. There are two types of co-occurrences: nouns occurring directly together, that is, in adjacent position (NN) and nouns whose co-occurrence is mediated via another type of word occurring in between them (possibly a preposition, adjective, verbs). Both types need to be identified. Nouns can be easily extracted, regardless of their distance to each other and regardless of the type and the number of words in between them, provided that none of them is a noun. The procedure works as follows: starting from the current noun, we increase the window size to the point to include the next noun. Having two nouns (car-engine; engine of car), we signal their respective functions via names, calling the first one the *head* and the second the *tail*. 'Car' and 'engine' are respectively the *head* and the *tail* in the 'car-engine' co-occurrence, while they are the reverse in the 'engine-car' example. Hence, cases where the *part* appears both before and after the *whole object* will be retrieved. Since the conclusion that a noun assumes the role of the part or whole may be incorrect, we have decided to delay this decision until the very end.

- *Step 3*: N-N co-occurrences with an identical tail noun take N-N co-occurrences from the preceding step to cluster them on the basis of their tail noun. For example, ‘corolla car' and 'door of car' belong to one cluster, both of them having the same *tail* noun: 'car', while 'car design' and 'network design' belong to another cluster. The same holds true for 'airplane engine,' 'search engine' and 'car engine'.

  car [corolla, door],
  design [car, network],
  engine [airplane, search, car]

- *Step 4*: the noun pairs of the clusters created in step three are clustered again, but this time on the basis of the similarity value of their head nouns.

  car { [corolla] [door] }
  design { [car] [network] }
  engine {[airplane, car] [search]}

  The similarity value is calculated by taking the cosine value of the vectors of the head nouns. The vectors are created by taking every word co-occurring with the noun (n-gram). This component and the next one require n-gram information. We got this from COHA[18]. All words are represented as a vector of their bi-gram value. Hence, each word has an n-gram value, represented as a vector. In order to calculate the similarity between the *head* nouns we used the cosine value of the vectors of the *head* noun. *Head* nouns whose cosine values are above a certain threshold are clustered together.
- *Step 5*: This component computes the similarity between the head and tail noun. In this module two types of similarity values are calculated. We call them S_1 and S_2. Note that the vector used to create S_1 in this module is different from the one used in the preceding step. The vector for S_1 is built here only on the basis of words co-occurring with the tail noun. If ever a word co-occurs both with the tail and the head noun, its n-gram value is recorded in both vectors, otherwise their respective vector values will be 1 for the tail noun and zero for the head noun. Words co-occurring only with the head noun will not be included in the vector. Hence, the size of the vector is equal to the size of the number of words co-occurring with the tail noun. However, in order to create a vector for S_2, we will also consider words co-occurring with the head noun also. The similarity value for S_1 and S_2 is again derived from the distance between the vectors i.e. their cosine value. The basic idea is that the *tail* nouns of the noun pairs presenting the 'Component-Integral object' or a 'Part-Whole relation' have a strong similarity value with their head nouns in their clusters. Hence, words like 'airplane' and 'car' have a strong similarity value with respect to 'engine', while 'search' has only a small one in the cluster: airplane-engine, 'search-engine', 'car-engine'.
- *Step 6*: the last module identifies whether two nouns are linked via an integral component Part-Whole relation or not (PT-WHR). To do so, the system draws on information provided by the above-mentioned modules. Given some cluster(s) (built in step 3 and 4) and a set of similarity values (identified in the training corpus, step 5), the system extracts automatically a production rule: if <condition> then <action>. This latter is used to decide whether two words are linked via an integral component PT-WHR or not. In order to achieve this goal, we took a corpus and tagged as “T” nouns pairs exhibiting a part-of

[18] http://www.ngrams.info

relationship and as "F" in the remaining cases. The system counts then the similarity values exhibited by the majority of noun pairs in the training set. The range of these values are learned automatically. The system calculates two similarity values (S_1, S_2) for every noun co-occurrence in the training set and takes then the range of values exhibited by the majority of part-of noun co-occurrences in the corpus. In order to determine this range, we calculated an error rate for all possible similarity ranges obtained for all NN co-occurrences in the corpus and selected the one with the lowest error rate. For example, suppose your corpus contained six NN occurrences (the first three being negative, the remaining being positive examples). Suppose further that the nouns having respectively the following values for S_1 (0.2, 0.3, 0.6, 0.8, 0.85, 0.9) and S_2 (0.1,0.3,0.4,0.45,0.5,0.55). This would yield the following result:

Range	% of negative relations retrieved	% of positive relations excluded
$S_1 < 0.2$ and $S_2 < 0.1$	0%	100%
$S_1 < 0.2$ and $S_2 > 0.1$	0%	100%
$S_1 < 0.2$ and $0.3 > S_2 > 0.1$	0%	100%
$S_1 < 0.2$ and $S_2 < 0.3$	0%	100%
$S_1 < 0.2$ and $S_2 > 0.3$	0%	100%
$S_1 < 0.2$ and $0.4 > S_2 < 0.3$	0%	100%
...	...	...
$S_1 > 0.2$ and $S_2 < 0.4$	100%	0%
...	...	...
$S_1 > 0.8$ and $S_2 > 0.4$	0%	0% (best range)

Table 1: samples of the possible ranges of similarity values generated and their error rate

We assume in the example above that the values of S_1 and S_2 of the first three lines are based on negative examples, while the remainder are positive, i.e. they contain a part of relation. In our case, most of the similarity values exceed 0.4 for S_1 and 0.8 for S_2. Here below is a subset of the algorithm: Given a pair of nouns as described in the steps 3 and 4 here above.

```
If the similarity value S2 > 0.4 && if the similarity value S1 > 0.8
   If the noun pairs occurred at least once as compound noun
   Then the head noun refers to the whole and the tail to the part
Else
   If the average similarity value (C) between the noun and the other nouns in the
     cluster > 0.4
   If one of the nouns in the cluster has S2 > 0.4 and S1 > 0.8
      If the noun pairs occurred at least once as compound noun
      Then the head noun refers to the whole and the tail to the part
   Else
      The relationship between the nouns is other than a whole-part relation
```

The rule stipulating that ‘noun pairs occurring at least once as compound noun’, does not imply that the noun referring to the 'part' is always the second noun, and the 'whole' the first. Indeed, the two may be separated by words of another type, for example, a preposition. In this case the arguments will swap position, the 'part' preceding the 'whole'. Both cases will be handled as discussed in step 2. Having extracting the nouns for both cases, we can find the pairs as a compound noun at least once in a well-balanced corpus. For example, ‘engine of car’ can be extracted as explained already in step 2, and the system will then interpret the pair as 'part-whole' if it exists as ‘car engine’, which is always the case in a well-balanced English corpus.

We managed to extract the specific semantic similarity patterns for NN co-occurrences exhibiting a part of relation. We also showed that different types of similarity measures (S_1, S_2) can be extracted from n-gram information. For example, for part_of relations we have extracted two types of similarity values (S_1, S_2) with their respective range of values. N-N co-occurrences that do not fall within the defined range are filtered out. They do not express part_of relations. Note that, unlike other approaches including LSA, we do not simply measure the similarity values of the two noun pairs, but we build two types of vectors to determine two similarity values (S_1 and S_2) and check them then according to a set of rules. Note also, that our similarity measures filter only part of relations, hence different measures will be required if we want to deal with other types of semantic relations.

The vectors used by us for identifying the similarity values are built automatically by the system. However, the way of developing a specific vector for encoding part-of relations is not based on learning from a training set, it is based on a set of observations and assumptions.

Words co-occurring with *parts*, say 'engine', will very frequently be the very object of which they are part (“car-engine, airplane-engine”.), but not vice versa. The two sets are quite different. While a ‘car’ may contain many parts ('tyre', ’steering-wheel', 'gear box', etc.)’, it may nevertheless be linked to many concepts playing another role than being a *part* : 'driver', 'accident', 'race', etc. Put differently, the link can be other than 'part_of'. Nevertheless, objects expressing a part are nearly always connected to the entity of which they are part of.

The example here below illustrates the functioning of the algorithm: at step 2 the algorithm lists N-N occurrences like car-engine, train-engine, airplane-engine, benzine-engine, gasoline-engine, and search-engine. N-N occurrences are put in the same cluster as they have the same tail noun : engine (step 3). In step 4 the cluster is further classified in to three sub-clusters: cluster $_1$, cluster $_2$ and cluster $_3$:

Cluster 1: VEHICLES [car-engine, train-engine, airplane-engine]
Comment: We have an integral component Part-Whole relation, as 'engine' is part of a holistic entity: VEHICLES (car, train, and airplane).

Cluster 2: OIL [benzine-engine, gasoline-engine]
Comment: 'Engine' is not part of 'oil' (benzine or gasoline).

Cluster 3: SEARCH-ENGINE

The two clusters here above are created within a cluster having engine as tail noun. The clusters are identified on the basis of the similarity value of the head nouns. Since 'car', 'train', and 'airplane' have a strong similarity value they are put in the same cluster. Likewise, 'benzine' and 'gasoline' are put into some cluster and so does 'search'. At step 6 the system separates the cluster

1 from the rest, as the vector similarity of 'engine' and 'oil' on one hand and 'search' on the other is below a given threshold value, while the one of 'engine' and 'vehicle' is above it.

5.3.1 A walkthrough

Let us explain our approach in more detail via an example. Suppose the following input :

> The Japanese government decided to raise taxes for the export of Toyota cars. This is not the only problem Toyota had to face during the last few months. Indeed, the motors of their new car models having problems, the company decided to revise for free all the recently cars sold.....

The POS tagger identifies in step-1 the part of the speech of the words

> The Japanese government decided to raise taxes for the export of Toyota/NP_1 cars/NN. This is not the only problem Toyota/NP_1 had to face recently. Indeed, the motors/NN of their new car/NN models/NN_2 having problems, the company/NN decided to revise for free all the recently cars/NN sold.....

At the next step we extract NN co-occurences: `Toyota-car; motors-car, car-models`, etc. At step-3 we cluster these co-occurences according to their tail noun :{[Toyota-car, motors-car], car models]]} At step-4, the head nouns are clustered according to their similarity value. This latter is based on the distance between the vectors of the head nouns (the nouns appearing first). This yields the following results: `Toyota`, `motors` and `car`. We also calculate at step-4 the dot product (similarity of the vectors of the head nouns).To create the vectors we use the N-gram information contained in the COHA corpus, that is, we take all words co-occuring with nouns. Words with similar vectors will be grouped in the same cluster. At step-5, we identify the similarity values (S_1 and S_2) for the head and the tail noun as shown in the table below:

NN co-ocurrence	S_1 for head	S_1 for tail	S_2
Toyota-car	0.73521462209380772	0.1348399724926484	0.099136319419321925
Motor-car	0.82118460785425675	0.519575448720232	0.40259135545057436

This is the way how vectors are built:

- The vector value is 1 for words co-occuring with 'Toyota' and 0 for words that, while not co-occuring with 'Toyota', do occur with 'car'. This allows us to create the vector S_1 for 'Toyota'. The S_1 similarity value for 'Toyota' is calculated by taking the distance (dot product) between the S_1 vector of 'Toyota' and a vector built on the basis of words co-occuring with both nouns (the intersection of 'Toyota' and 'car'). Put differently, the vector is built by taking words whose similarity value is 1 in both vectors, for example, 'Toyota' and 'car'.
- Likewise, the vector value is 1 for words co-occuring with 'car' and 0 for words, that while not co-occuring with 'car' do co-occur with 'Toyota'. This allows us to build the S_1 for 'car'. The S_1 similarity values for 'car' are calculated by taking the distance (dot product) between the S_1 vector for 'car' and a vector built on the basis of words co-occuring with

both nouns (the intersection of 'Toyota' and 'car'). As here above, the vector is built by taking words whose similarity value is 1 in both vectors (again, 'Toyota' and 'car').

- The S_2 similarity value is calculated by taking the dot product between the S_1 vectors of 'Toyota' and 'car'.

How do we decide wheter a relationship is of the kind 'part_whole' (step-6)?

The rules use the similarity values of the table here above in order to decide whether there is a meronymic relation between the two nouns, and what respective roles of the nouns are (which is the 'whole' and which is the 'part'). This is how the rule works. S_1 is 0.73521462209380772 for 'Toyota' and 0.1348399724926484 for 'car', S_2 being 0.099136319419321925. Likewise, S_1 is 0.82118460785425675 for 'motor' and 0.519575448720232 for 'car', the value of S_2 being 0.40259135545057436.

Assume that N_1 and N_2 are respectively the first and the second noun. Hence, N_1 S_1 and N_2 S_1 are the respective S_1 similarity values of the first and the second noun, S_2 being identical for both nouns. The production rule checks now the similarity values against the treshold learned from the training set, the thresholds being the ranges of the similarity values exhibited by most of the meronyms in the training set.

if (N_1S_1 > = 0.8 and S_2 > = 0.4) then print: N_1 <part>; N_2 <whole>

if (N_2 S_2> = 0.8 and S_2 > = 0.4) then print: N_2 <part>; N_1 <whole>

In the 'Toyota-car' co-occurrence, 'Toyota' and 'car' are respectively N_1 and N_2. N_1S_1 is S_1 for 'Toyota', while N_2S_2 is S_1 for 'car'. Substituting the values in the rule would yield:

if (0.735 > = 0.8 and 0.099 > = 0.4) then print ('Toyota' is <part> and 'car' is <whole>)

if(0.135 > = 0.8 and 0.099 > = 0.4) then print ('car' is the <part> and 'Toyota' is the <whole>)

Since none of the above apply, the relationship between the nouns is other than a meronymic one. Let's do the same for 'motor-car':

if (0.821 > = 0.8 and 0.402 > = 0.4) then print ('motor' is the <part> and 'car' is the <whole>)

The condition stated in the rule is satisfied by the similarity value of the noun pairs. Hence, we do have a meronymic relationship with 'motor' being the <part> and 'car' being the <whole>.

if (0.51 > = 0.8 and 0.402 > = 0.4) then print ('car' is the <part> and 'motor' is the <whole>), which is false.

The steps just described are performed for all NN co-occurences in the paragraph.

5.3.2 Identification of the links senses

The concepts and the links holding between them are thus extracted from the corpus as explained above. However, there is one other problem that needs to be addressed. A word may express several meanings. For example, the word-form (lemma) 'mouse' may stand for a 'rodent' (animal) or a 'computer device'.

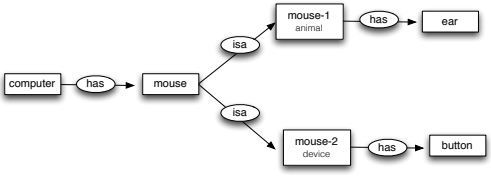

FIGURE 3- Sample of the semantic map for two senses

Likewise, the noun 'table' has various senses. WN[19] lists among others the following four:

- S1 (n) table, tabular array (a set of data arranged in rows and columns). Example: 'mathematical table'
- S2 (n) table (a piece of furniture having a smooth flat top that is usually supported by one or more vertical legs). Example : 'it was a sturdy table'
- S3 (n) table (a piece of furniture with tableware for a meal laid out on it). Example: "'I reserved a table at my favorite restaurant'
- S4 (n) table (a company of people assembled at a table for a meal or game). Example: 'he entertained the whole table with his witty remarks'

Of course, we have to identify (possibly automatically) which one of them applies in our case, as different senses, say 'array' rather than 'kitchen table', encode different semantic relations and arguments (['row' and 'column'] vs. ['leg', 'tabletop', 'meal' and 'tableware']).

In order to identify the senses, we start by listing all the parts of the concepts and cluster then the extracted parts on the basis of the cosine value between their vectors constructed from their n-gram. Polysemous words, that is concepts/words with several senses, will have several clusters. The links/associations holding between the concepts are marked on the basis of their senses. Hence, the link between two concepts encodes two types of information: the nature of the semantic relationship and the sense. In our current version we have only one type of relation i.e. meronym and the senses are not labelled semantically.

The senses are learned from the number of clusters built on the basis of the parts of the concepts. Example, 'table has parts: column, row, leg, tabletop and tableware'. The cosine value of each part is compared with all other parts to identify the clusters. To this end we used the k-means clustering technique[20]. In our 'table' example, 'column and row' and 'leg, tabletop and tableware' are grouped together given their respective vectors.

To identify senses we use like (Rapp, 2004; Diab & Resnik, 2002; Kaji, 2003; Pinto et al., 2007) a clustering method. However, our task is narrower in that the clusters are formed only from a small set of words associated with a given word at a time. Also we have considered meronymic word senses only i.e. senses that affect PT-WHRs.

The extracted wholes and their parts are organized into a network. Concepts are organized hierarchically i.e. going from the whole to its parts. For example 'tooth' is part of 'gear' which is part of an 'engine' which is part of a 'car'. In this case, 'car' is the root. Concepts which are parts of

[19] http://poets.notredame.ac.jp/cgi-bin/wn

[20] http://en.wikipedia.org/wiki/K-means_clustering

several concepts are connected via several links. For example, 'engine' being part both of 'car' and 'train' it has two incoming links (see figure 4)

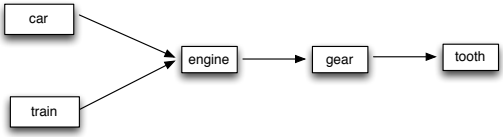

FIGURE 4-Sample of the semantic map showing multiple links

5.4 Evaluation

We have tested our system for its ability to extract PT-WHRs by using the text collection of SemEval (Girju et al. 2007). The test corpus is POS-tagged and annotated in terms of WN senses. The corpus has positive and negative semantic relations. The corpus has positive and negative semantic relations. The part–whole relations extracted by the system were validated by comparing them with the valid relations labeled in the test set answer key. The format of the test set is described in the sample here below:

"Some sophisticated <e2>tables</e2> have three <e1>legs</e1>."
WordNet(e1) = "n3", WordNet(e2)="n2"; Part-Whole(e1, e2) = "true"

This format has been defined by Girju et al (Girju et al. 2007). Since this does not correspond to a real text format, we have changed the corpus accordingly, to obtain the following text: "Some sophisticated tables have three legs". To evaluate the performance of our system we defined precision, recall, and F-measure metrics in the following way:

Recall	$\frac{\text{Number of correctly retrieved relations}}{\text{Number of correct relations}}$
Precision	$\frac{\text{Number of correctly retrieved relations}}{\text{Number of relations retrieved}}$
F-measure	$\frac{2}{\frac{1}{precision} + \frac{1}{recall}}$

Our system identified almost all (19/20) of the *present Component-Integral object part-whole* relation pairs of the SemEval test set. Since these relations are both present and non-present in the Semeval training set and test set, we considered the present relations to evaluate the performance of our approach.

As the number of concepts having parts in different senses is very small in the SemEval test set, we have added some concepts from WN. The resulting number of relation pairs accounts now for 20% of our test set. 80 % of this set contains negative examples coming either from the SemEval test set (all of them) or from our own. We defined 'recall' as the percentage of correctly retrieved relations out of the correct relations available in the test set, while 'precision' is defined as the percentage of correctly retrieved relation out of retrieved relations. We obtained 95,2% for precision, 95% for recall and 95,1% for the F-measure. The PT-WHRs extracted by the system

were validated by comparing them with the valid relations labeled in the test set answer key. The test set has answer key, so we manually counted correctly retrieved relations. Table 2 is a sample of correctly retrieved relations: Arm wrist, man head, hand finger, car engine. The following table shows the similarity values of some noun pairs taken from the program :

The noun pairs	S_1 similarity values	S_2 similarity value	interpretation
'car', 'engine'	0.8788321167883211	0.4524886877828054	part_of
'search', 'engine'	0.5040650406504065	0.3229166666666667	other
'chemistry', 'laboratory'	0.6666666666666666	0.28426395939086296	other
'laboratory', 'hand'	0.5238095238095238	0.06063947078280044	other
'hand', 'finger'	0.8631840796019901	0.49118457300275482	part_of
'arm wrist'	0.8911223341267891	0.59118958311003478	part_of
'man head'	0.8234512378001223	0.43407700124560945	part_of

Table 2: the similarity values of selected noun pairs

All the encountered errors are hyponyms ('car' and 'vehicle'). However, this does not imply that all the hyponyms in the test are incorrectly retrieved as part-whole relation. Actually, only 12% of the hyponyms in the test set are incorrectly retrieved as part-whole relation. It should also be noted that the majority (80%) of our test set relations are not *part-whole relations*. Therefore, the probability of randomly selecting *part-whole relation* is 20/80 (0.25), showing the effectiveness of this approach for discriminating such relations.

We have also evaluated the performance of the system in determining the senses of a concept. To do so we used the clustering technique described above. Word forms expressing several senses have several clusters. We evaluated the results against the gold standard of meronymic word senses taken from WN (Miller, 1990).

Our clustering is based on the distance between the vectors of the parts of a given concept. We defined precision as the percentage of words assigned to their actual WN meronymic senses out of total words assigned to output clusters. Recall is the ratio of words assigned to their actual WN meronymic senses' correct relations available in the test set. We have achieved 89% for precision, 86% for recall and 87, 47 % for the F-measure.

6 Conclusion

We have started this paper by arguing that relational information is important for many tasks. We were concerned here mainly with lexical access, a very important task in language production (speaking, writing). Noting that current dictionaries do not support authors as well as needed, —a criticism that holds even for electronic dictionaries despite the recent progress,— we suggested to add to an existing electronic resource an index based on the notion of associations, i.e associated words to a prime (source word) and relations holding between the two associated words.

Since this index is based on the co-occurences of words in a corpus, —the latter representing ideally the user's world-knowledge, and since this knowledge changes frequently, it is desirable to allow for updating the index dynamically by taking into account the changes of the corpus. Hence, the idea to extract the links or associations automatically. As this is a very complex problem, we decided to study its feasibility only for a small subset, meronymic relations.

Despite certain shortcomings (this is work in progress), the results obtained are quite promising. This is all the more encouraging as we used very few resources compared to similar works. We believe that this approach can be generalized, allowing us to extract other types of semantic relations. But of course, much more work is needed to substantiate this latter claim.

References

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

Beamer, B., Rozovskaya, A. and Girju, A. (2008). *Automatic Semantic Relation Extraction with Multiple Boundary Generation*. Association for the Advancement of Artificial Intelligence.

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

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

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

Dell, G. and Juliano, C. (1996). Computational models of phonological encoding. T. Dijkstra et K. De Smedt (Eds.), Computational Psycholinguistics. London: Taylor & Francis, 328-359

Diab, M. and Resnik, P. (2002). *An unsupervised method for word sense tagging using parallel corpora*. In Proc. of ACL.

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

Ferret, O. and Zock, M. (2006). *Enhancing Electronic Dictionaries with an Index Based on Associations*. 21st Intern. Conference on Computational Linguistics, Sidney

Finin, T. (1980). *The semantic interpretation of compound nominals*. Ph.D. Dissertation, University of Illinois at Urbana-Champaign.

Firth, J.R. (1957). *A synopsis of linguistic theory* 1930-1955. In Studies in Linguistic Analysis, pp. 1-32. Oxford: Philological Society.

Galton, F. (1880). *Psychometric experiments*. Brain, 2, 149-162.

Girju R., Moldovan D., Tatu, M. and Antohe, D. (2005). *Automatic Discovery of Part–Whole Relations*. ACM 32(1)

Girju, R., Hearst, M., Nakov, P., Nastase, V., Szpakowicz; S., Turney, P. and Yuret D. (2007). *Classification of semantic Relations between Nominals: Dataset for Task 4 in SemEval*, 4th International Workshop on Semantic Evaluations, Prague, Czech Republic.

Hage, W., Kolb, H. and Schreiber, G. (2006). *A Method for Learning Part-Whole Relations*. TNO Science & Industry Delft, Vrije Universiteit Amsterdam.

Harris, Z. (1954). *Distributional structure*. Word 10 (23), 46–162.

Harshman, R. (1970). *Foundations of the parafac procedure: Models and conditions for an "explanatory" multi-modal factor analysis*. UCLA Working Papers in Phonetics, 16.

Hearst M. A. (1998). *Automated Discovery of WordNet Relations*. In Fellbaum, C. (Ed.) WordNet: An Electronic Lexical Database and Some of its Applications, MIT Press. pp. 131-151

Kaji, H. (2003). *Word sense acquisition from bilingual comparable corpora*. In Proceedings of NAACL.

Lafourcade, M. (2007). *Making people play for Lexical Acquisition with the JeuxDeMots prototype.* In 7th International Symposium on Natural Language Processing, Pattaya, Chonburi, Thailand.

Landauer, T.K., & Dumais, S. (1997). *A Solution to Plato's Problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge*. Psychological Review, 104, 211-240.

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

Lund, K. and Burgess, C. (1996). *Producing high-dimensional semantic spaces from lexical co-occurrence*. Behavior Research Methods, Instruments, and Computers, 28(2), 203–208.

Mark, D. (2011). *N-grams and word frequency data from the Corpus of Historical American English (COHA)*.

Matthew, B. and Charniak, E. (1999). *Finding parts in very large corpora*. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, pp. 57–64, University of Maryland.

Miller, G. (ed.) (1990). *WordNet: An On-Line Lexical Data-base*. International Journal of Lexicography, 3(4), 235-312.

Nelson, D. L., McEvoy, C. L., & Schreiber, T. A. (1998). *The University of South Florida word association, rhyme, and word fragment norms*. http://www.usf.edu/FreeAssociation/

Pinto, D., Rosso, P. and Jimenez-Salazar, H. (2007). *Word sense induction using self-term expansion.* In Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), 430–433.

Rapp, R. (2003). *Word sense discovery based on sense descriptor dissimilarity*. In: Proceedings of the Ninth Machine Translation Summit, New Orleans, pp. 315–322.

Rapp, R. (2004). *A practical solution to the problem of automatic word sense induction*. In proceedings of the ACL 2004 on Interactive poster and demonstration sessions.

Salton, G., Wong, A. and Yang, C.-S. (1975). *A vector space model for automatic indexing.* Communications of the ACM, 18(11), 613–620.

Schank, R. (ed.) (1975). *Conceptual Information Processing*, New York, American Elsevier.

Schvaneveldt, R. editor. (1989). *Pathfinder Associative Networks: studies in knowledge organization*. Norwood. N.J.

Sowa, J. (1992) *Semantic networks*. In 'Encyclopedia of Artificial Intelligence', edited by S. C. Shapiro, Wiley, New York

Vieu, L. and Aurnague, M. (2007). *Part-of relations, functionality and dependence*. In Aurnague, M., Hickmann, M. and Vieu, L. (eds.), The Categorization of Spatial Entities in Language and Cognition, pp. 307–336. J. Benjamins, Amsterdam

Wierzbicka, A. (1996). *Semantics*: *Primes and Universals*. Oxford University Press, Oxford.

Winston, M., Chaffin, R. and Hermann, D. (1987). *Taxonomy of part-whole relations*. Cognitive Science, 11(4), 417–444.

Wittgenstein, L. (1922). *Tractatus Logico-Philosophicus*. London: Routledge & Kegan Paul

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

Zock, M., Wandmacher, T. and Ovchinnikova, E. (2009). *Are vector-based approaches a feasible solution to the 'tip-of-the-tongue' problem*? Granger S. & Paquot, M. (Eds.) eLexicography in the 21st century: New challenges, new applications, Louvain-la-Neuve. pp. 355-366