

WordSets: Finding Lexically Similar Words for Second Language Acquisition

Vera Sheinman
Department of Computer Science
Tokyo Institute of Technology, Japan
vera46@cl.cs.titech.ac.jp

Takenobu Tokunaga
Department of Computer Science
Tokyo Institute of Technology, Japan
take@cl.cs.titech.ac.jp

Abstract

We introduce a method of expanding a multiple-words input by a short list of similar words in a manner suitable for Second Language Acquisition (SLA). Similarity for that purpose is determined based on two aspects, semantic relations and typicality. Finding words with similar typicality is particularly important for SLA tasks. The study incorporates, and shows the advantage of a recently introduced distance measure that uses the Web as its corpus. The value of the proposed method is demonstrated by empirical experiments on word lists provided by teachers.

Keywords

Second language acquisition, lexical acquisition, similar words, typicality, familiarity, similarity.

1. Introduction

Computational modeling of Second Language Acquisition (SLA) may be a great step toward a deeper understanding of how humans acquire new languages. Rappoport and Sheinman in [14] proposed a preliminary computational model of SLA. One of the components of their model is the prior conceptual knowledge of the learner. Existence of such knowledge is one of the major differences between SLA and First Language Acquisition (FLA). Hence, it requires special attention in SLA studies. In their study that component was constructed manually and was tailored to a specific corpus. A construction of an extensive model of learners' conceptual system is important. Ontology is one of the ways to do so, reflecting the recent beliefs about the structure of conceptual knowledge in psycholinguistic research. WordNet [12] may be viewed as one of the most extensive ontologies of that kind available.

This study introduces a method to compute conceptual categories, based on several examples. Proposed method will allow for (semi)automatic construction of an adult learner's conceptual system model. Additionally, this method may be applied as a tool for language courseware authoring, as well as a helpful tool for language learners, or even native speakers that are missing a word. For instance, if there is a difficulty retrieving the word for 'kiwi', entering examples of similar fruits such as 'apple' and 'lemon' might be a way to retrieve the missing word.

The type of learning that we analyze for the purpose of this study is generalization from examples, similar to [14]. After the learner hears enough examples in the second language, he is ready to generalize into a construction and he is able to generate new phrases. Learners are unlikely to generalize after a single example. In our study we require an input of at least two words to trigger recognition of a conceptual category and automatic extension of it. The scope of the current study is English nouns.

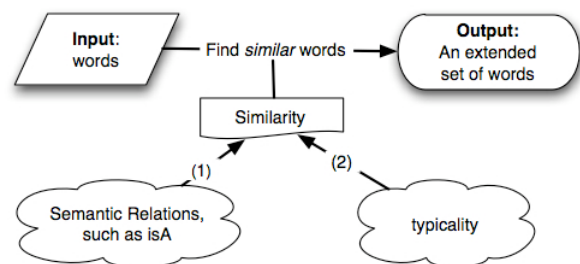


Figure 1: Diagram of Problem Definition

2. Problem Definition

A sketch of the problem that we suggest to solve automatically in the current study is shown in Figure 1. The 'WordSets' method, and an application implementing it, are the key products of this study. As part of the solution to this problem, we define 'similarity suitable for SLA tasks'. We focus on two aspects of similarity, described in the subsections below.

2.1 Semantic Relations

Words or concepts may be represented in an extensive network, such as WordNet, with many types of links connecting them. For instance, one such link is the 'isA' relation, or in terms of WordNet, the hyponym-hypernym relation. Focusing on two concepts out of the whole network reduces the numerous possibilities to consider to only the links that connect them. Choosing more than two concepts reduces the links even more, and provides further information about the similarity of these concepts.

The given input words share some semantic relations. We detect two such relations by looking for the least common subsumer of the given concepts, traversing the appropriate relation links in WordNet network.

2.2 Typicality

Some concepts are more common than others, while some are rare or even obscure. More common concepts are usually more likely to be encountered, and it is more important to learn the words representing them in the early stages of SLA. Typicality of the given words provides further information about the words and about the desired extension, and it should not be underestimated. If the given words share similar typicality, their most suitable extensions should share that typicality as well.

Consider the following example, in the context of a learner searching for an extension for a set of words he provides:

Input: olive, navy, maroon
Output: red, blue, yellow

The words provided in the input are not obvious choices for colors. Extending the set by the most basic colors will not provide the information that is probably being sought. In the context of a learner, if he knows such words as ‘maroon’, it is improbable that he does not know ‘blue’. The information in the output will be redundant for him.

In the opposite case, for very typical members of a category provided as the input, presenting complex words as the most similar extensions will overwhelm the learner. Moreover, it will not be useful for courseware authoring that seeks simple category members, easily recognized by students. Additionally, it will not be useful for modeling the core conceptual system of the most useful concepts based on typical examples.

3. Related Work

There is a large body of research and products that deal with finding similar words for a single entry. Additionally, there is an extensive body of work for measuring semantic similarity between two given words. Some of these studies base their similarity measures on WordNet [3]. Others exploit various computational techniques to measure such similarity in a corpus [5], explore psycholinguistic data, etc. One of the major directions is distributional similarity. An influential work by Lin [10] in this field analyzes syntactic features from a corpus, and comes up with rather broad clusters of similar words, synonyms and hyponyms mixed. Weeds and Weir [15] provide an excellent survey on distributional similarity techniques. It is still difficult to distinguish among the various semantic relations such as hyponyms or holonyms by these techniques, a knowledge that we need to protect the learners from unnecessary information.

Most previous studies refer to WordNet as the major available lexicon. Some previous studies on lexical similarity [6], [15], [16] use WordNet as the golden standard for evaluation purposes, especially for nouns. In this study, we focus mainly on ordering similar nouns by

typicality, using well-defined semantic relations, and hence we extract words similar to the input words directly from the ready constructed WordNet, using WordNet-based similarity measures. In this sense other studies on similarity are complementary to this study.

We work with an input of at least two entries, similarly to learners that generalize based on at least two examples. This task is essentially different from the task of finding similar items based on a single example, that most of the lexical acquisition works tackle.

The problem of providing similar items based on entry of several words may be viewed as **Ontology Learning** - provided existing entries in an existing category, this category is extended.

Although in reality some examples that learners encounter may be erroneous, they will still be able to create correct generalizations eventually. However, for the purpose of this study we compute the set of items that are equally similar to each one of the input entries, leaving possible inaccuracies or inconsistencies in the input set out of its scope.

Nation [13] recommends that language teachers avoid introducing words from lexical sets simultaneously. Some textbooks [18] follow this recommendation, and extend the lexical sets gradually. The research in this field is complementary to our work. Automatic construction of semantically related concepts might help teachers and textbook authors to be aware of such limitations.

3.1 GoogleSets

GoogleSets [7] is one of the projects in Google labs that provides a friendly tool to extend sets of words. Similarly to the proposed method, it receives multiple words as its input and provides an output of words similar to the input. GoogleSets is an efficient, dynamic, and generic application. It works for any kind of inputs (simple words, movie names, numbers, etc.), using the Web as its corpus.

Table 1. GoogleSets Results Example

GoogleSets Output (first 8 words) for Input: Doctor, Engineer
<i>Bureaucrat, Fixer, Enforcer, Trader, Adventurer, Soldier, Scientist</i>

However, lacking any specific linguistic objectives or any linguistic knowledge augmentation, it may not provide for building ontologies of conceptual systems of humans, or serving as a tool for learners. Table 1 shows an example of this idea. ‘Doctor’ and ‘Engineer’ are both very typical professions, and it is likely to assume that such similarly typical items as ‘Nurse’ or ‘Teacher’ are anticipated as the output. Instead, ‘Bureaucrat’, whose semantic similarity to the input set is questionable is the first word returned. Also, ‘Enforcer’, which is much less typical than the provided examples is one of the top results.

Although ‘*Soldier*’ and ‘*Scientist*’ come last on the extension list, they seem to be the best extensions.

Our method may be viewed as an adaptation of the GoogleSets results to make it suitable for SLA purposes.

3.2 Normalized Google Distance (NGD)

Cilibrasi and Vitanyi [5] introduce a distance measure between concepts, intended for large corpora such as the Web. Using the whole Web as the corpus, with the computational ease of acquiring page counts is a good method to obtain averaged information about what is typical and what is not. NGD is incorporated in our method for measuring typicality of words.

4. The Proposed Method

Given a set of words $W = \{w_1, \dots, w_n \mid n \geq 2\}$ (1) as the input, our method comprises 4 stages leading to output of a set of similar words. These 4 stages are described in the subsections below.

4.1 Disambiguation

In this stage, we perform word sense disambiguation (WSD) to determine the semantics of the words in (1). We assume that the words in (1) are similar enough, and consequently they can serve as the context for each other of the words in the set. The procedure is as follows.

Step 1: For each word w_i in W (1), acquire its noun senses $\{n_{i1}, n_{i2}, \dots\}$ from WordNet 2.1,

$$S = \{\{n_{11}, \dots, n_{1m}\}, \dots, \{n_{n1}, \dots, n_{nk}\}\}. \quad (2)$$

Step 2: For each combination of senses in (2), compute the sum of Lesk similarity measures [1] between its members pairwise.

Step 3: Determine the combination with the highest sum of similarities $SD = \{n_{1x}, \dots, n_{ny}\}$. (3)

There are several approaches for WSD task. In this study we search for semantic relation information, and it makes sense to use WordNet-based similarity measures to perform disambiguation.

Budanitsky and Hirst [3] in their thorough evaluative survey suggest that the measure by Jiang-Conrath [8] is superior to other WordNet-based measures. However, this measure does not provide any results for many entries. Additionally, although this measure is very effective in measuring similarity between entries that share the same hypernym in WordNet hierarchy, it is not as effective for entries that are similar by other relations, such as meronymy. As opposed to Jiang-Conrath, Lesk measure that is based on gloss overlaps in WordNet reflects similarity between words with meronymy relation equally well. Recent studies [11] report on Lesk outperformance of Jiang-Conrath for the purposes of WSD.

The meronymy relation is important for our task where the input words often tend to be parts (meronyms)

of some concept. For instance, the words ‘*bumper*’ and ‘*window*’ that are both meronyms of ‘*car*’ cannot be disambiguated by Jiang-Conrath. However, Lesk provides a correct disambiguation for them.

4.2 Detection of Semantic Relations

We assume that the word senses in (3) share some semantic relations. Two shared relations may be detected automatically using WordNet relations:

$$\begin{aligned} Z_1 &= \text{least_common_holonym_in}(SD), \\ Z_2 &= \text{least_common_hypernym_in}(SD), \quad (4) \\ R &= \{\text{meronyms}(Z_1), \text{hyponyms}(Z_2)\}. \end{aligned}$$

Z_1 in (4) may be non-existent, due to the structure of WordNet. For instance, $\text{apple}\#\text{n}\#1$ ¹ and $\text{pear}\#\text{n}\#1$ do not share a holonym. In such case the relation $\text{meronyms}(Z_1) = \phi$. Z_2 , however, always exists.

4.3 Extension

In this stage the set of word senses SD (3) is extended by adding the word senses that are acquired by recursive WordNet traversal for each of the relations in R (4),

$$\begin{aligned} E_1 &= \{n_{1x}, \dots, n_{ny}, e_{11}, \dots, e_{1m}\}, \\ E_2 &= \{n_{1x}, \dots, n_{ny}, e_{21}, \dots, e_{2h}\}. \end{aligned} \quad (5)$$

The items that are deeper by more than one level than the deepest item in the input in the WN hierarchy are not added to the input. This is done, in order to prevent overly specific items, or instances appearing in the same lexical set with other items. For example, consider ‘*airport*’ and ‘*bank*’ provided as an input. In the context of extraction of words from examples, the user might expect to see ‘*hospital*’, or ‘*gas station*’ as other examples of institutions, rather than ‘*Kennedy airport*’ or ‘*Mutual Savings Bank*’ that are of greater specificity than the items in the input.

For the simplicity of calculation, we remove relations that have a very general hypernym, such as ‘object’ or ‘substance’. We determine the intended extension as too general when Z_2 is closer to the WordNet root than to the items in the input, so that $\min_{SD} (\text{depth}(n_{ij}) - \text{depth}(Z_2)) > 2/3 (\text{depth}(Z_2))$.

The pruning techniques mentioned above will malfunction in certain cases, due to the unbalanced state of WordNet hierarchy. Better methods will be considered in the future studies.

4.4 Ranking Procedure

The suggested ranking procedure is the key part of our study. It is counter-productive to overwhelm learners with information. Ranking the results will allow us to

¹ The notation $\text{apple}\#\text{n}\#1$ stands for the first noun sense for the word ‘apple’ in WordNet. It refers to the fruit ‘apple’.

differentiate between the more useful and less useful extensions of the given set.

Given the extended sets of word senses (5), the elements of each set will be ranked by their typicality (section 2.2). The items with typicality level closest to the input words will be ranked the highest. The Web is a huge corpus, with plethora of domains evening out the typical usages. We use frequencies in the Web as the markers for typicality.

In order to calculate typicality we use the distance measure of NGD (section 3.2). NGD requires M (the total number of pages indexed by a search engine). Most of the large search engines do not declare this number. We estimate M by retrieving the number of webpages that include the word ‘the’, and restrict the search to English pages. An interesting study [2], suggests an improvement for this kind of estimation. We plan to experiment with the suggested measure in the future.

An interesting feature of NGD is that it tends to cluster items not only by their similarity, but also by their frequency. For instance, the colors ‘red’, and ‘blue’ are clustered together, apart from ‘pink’, and ‘wine’, which seem more similar to ‘red’ than ‘blue’ [4].

NGD measures the distance between two items - x , y . We measure the distance between a set of items to one item - X , y . For the purpose of this study we used the

$$distance(X, y) = \sum (NGD(x, y) \mid x \in X). \quad (6)$$

The smaller the distance of an item from the input set, the higher its ranking.

When submitting queries to a search engine, we once again use words, rather than WordNet senses. Hence, we need further disambiguation, in order to prevent many results such as “Apple computer” biasing our calculation when dealing with an input of ‘apple’ and ‘pear’. This is achieved by incorporation of NGD. Similarity is measured between each input word and the word in question. We implement the distance measure using estimated counts by Yahoo.

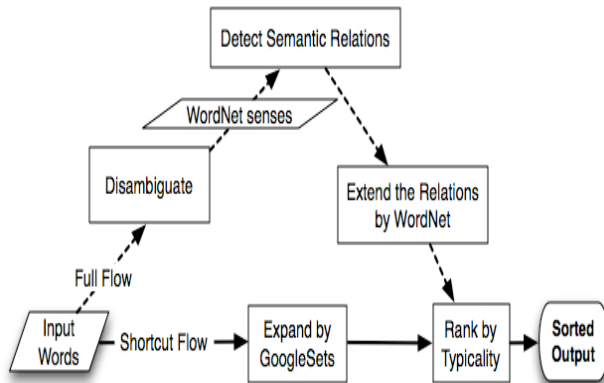


Figure 2: Two possible flows for 'WordSets'

5. Shortcut Flow

The main focus of this study is on the ranking of words by their similarity to the words provided in the input. In order to evaluate only this stage, and also in order to provide solution for the cases when WordNet does not include the input words, we introduce an alternative shortcut flow. The two possible flows in general are overviewed in Figure 2. The steps of the shortcut flow are presented below.

Step 1: Expand the input words by the larger set in GoogleSets.

Step 2: Standardize the results, due to inconsistency of GoogleSets results in terms of capital letters and such. This step is performed using the validity check provided in WordNet. All the nouns are stored in their singular form in low-case letters for consistency.

Step 3: Rank the results by the same ranking procedure as described in section 4.4.

Step 4: Output the results sorted by their ranking.

6. Evaluation

In order to test our method, we have performed several evaluation procedures as described in the subsections below.

Table 2. The evaluation of the full flow using WordNet

Word lists	Precision% full / reduced	Recall% full / reduced
Family	8 / 49	76 / 49
Colors	9 / 78	83 / 78
Vegetables	11 / 33	81 / 33
Buildings	0 / 0	0 / 0
Fruits	3 / 27	30 / 27
Clothes	5 / 21	47 / 21
House	4 / 7	19 / 7
Tools	3 / 50	88 / 50
Body	4 / 18	34 / 18
Animals	2 / 6	12 / 6
Macro average	5 / 29	47 / 29
Micro average	6 / 36	54 / 36

6.1 Lexical Sets from Word Lists

Ten lexical sets were retrieved from word lists provided by English teachers for beginners [9] from a site for English learners in Japan. For each one of the lexical sets two of its members were randomly chosen as the input words. The rest of the words served as test set. Both, the full procedure using WordNet (section 4), and the shortcut procedure (section 5) were performed for at least two different input sets for each word list. In total 32 different input sets were tested, and 32 hyponyms and 5 meronyms relations were detected. In cases when the size of the acquired set was big enough the set was reduced to

the same size as the appropriate word list size after sorting it by ranks. We compared the precision rates for the full set (before ranking) vs. the reduced set (after ranking).

Table 3. Shortcut flow evaluation

Comparison of our method (WS) with GoogleSets (GS)				
Word lists	Precision % full / reduced		Recall % full / reduced	
	GS	WS	GS	WS
Family	41 ² / 61	42 / 65	82 / 56	82 / 59
Colors	24 / 89	24 / 69	100 / 89	100 / 69
Vegetables	29 / 47	36 / 56	67 / 47	79 / 56
Buildings	8 / 8	9 / 11	9 / 8	11 / 9
Fruits	44 / 56	48 / 53	100 / 56	100 / 53
Clothes	28 / 37	23 / 33	34 / 30	26 / 26
House	3 / 4	4 / 5	2 / 2	3 / 3
Tools	8 / 25	8 / 38	44 / 25	44 / 38
Body	43 / 52	33 / 38	66 / 52	46 / 38
Animals	56 / 66	62 / 69	51 / 31	53 / 30
Macro avg.	28 / 44	29 / 44	59 / 39	57 / 38
Micro avg.	29 / 47	30 / 47	63 / 43	64 / 43

To illustrate the evaluation process consider the word list for 'tools' that contains 10 words: *drill, hammer, knife, plane, pliers, saw, scissors, screwdriver, vise, and wrench*. Two input word pairs were randomly chosen 'drill, pliers', and 'hammer, vise'. For the first input set, 228 words were extracted from WordNet, and 43 words were extracted from GoogleSets. Precision and recall values were first calculated for these lists comparing them to the original word list of tools. As the next step we sorted both of the lists by our ranking procedure and reduced each of the sets to the first 10 words. Then, we recalculated precision and recall for the shorter lists to evaluate our ranking procedure's contribution. For comparison of the sorting we also reduced the list by GoogleSets in the same manner, without ranking it. The same procedure was performed for the second input set.

Our main purpose in the analysis is to show improvement of precision for the reduced ranked lists. Perfect precision values cannot be anticipated, because the chosen lists are a sample of word lists that typically appear in textbooks. They may omit some words, due to size limitations or other reasons. However, improvement of precision after ranking shows good tendency toward conformity with the teachers' opinions. Recall values are expected to decrease due to reduction of the acquired sets. The precision values for the full procedure that are shown in Table 2 clearly suggest that the ranking procedure

successfully cleans the word sets from redundant items, increasing the precision by 6 times on average for each list. The best ranking was achieved for *colors* with inputs 'orange, white', 'black yellow', and 'green, purple'.³

The precision results for the ranking procedure in comparison with GoogleSets show similar values on average (see Table 3). Precision in this experiment is higher than in the full flow (see Table 2), due to better order by similarity and typicality of items in GoogleSets, compared to non-existent order in WordNet synsets. Note the better precision and recall for the ranked *tools* set with inputs 'drill, pliers' and 'hammer, vise'. Ranked lists show better results for 6 word lists, and worse precision for *colors, fruits, clothes* and *body parts*.

6.2 Familiarity Rating

Familiarity values used for this experiment were extracted from the MRC Psycholinguistic Database [17]. The total number of rated words extracted was 4896, from the lowest rating of 101, to the highest of 657.

All the words (total of 19) in the category of 'vegetables' that appear both in WordNet and in familiarity rating were extracted. One copy of the list, noted by *F*, was sorted according to its familiarity rates, another copy *X* was ranked using the ranking procedure as described in section 4.4 using the top two familiar items from *F* as the input. The order of the two lists was compared summing the absolute error as following.

$$rank_L(x) = \text{the position of item } x \text{ in list } L$$

$$error(X) = \sum |rank_F(x) - rank_X(x)|$$

The error for the ranked set is 48 and the mean error (calculated combinatorically) is 96. The order of the ranked set is two times more similar to the list *F* than the average. Discrepancies in the order of the sets are anticipated. One of the contributions to the inconsistency may be relatively old dating of the familiarity rating experiments. The typicality ratings are based on a more recent language that appears in the web.

7. Discussion

We have pointed out the needs of SLA in the field of computerized lexical acquisition. Motivated by them, we have divided the former known notion of similarity into two aspects of semantic similarity and typicality level similarity, and we have presented a method for semisupervised lexical acquisition from multiple words input based on this new notion. Our method is web-based, hence, providing dynamic results that reflect the changes that happen in the language use from day to day.

² The precision values for GS and WS before reduction, sometimes differ due to the standartization procedures applied on GoogleSets result before ranking it (step 2 in section 5)

³ In some cases, the results acquired from WordNet were too general, or there were errors in the disambiguation. In such cases, we reran the tests with additional input words.

We implemented the suggested method using the distance measure of NGD, and compared it to the existing application of GoogleSets. NGD is a universal measure that measures distance over all the implicit similarity aspects between two items. It does not require an annotated or parsed corpus. We have shown its applicability to the similarity by typicality level. We plan to compare its usefulness with additional approaches and similarity measures in the future.

Integration of the presented method into computational modeling of SLA seems to be a much needed direction. Additionally to the theoretical value, being able to extend several example words by words of similar typicality and semantic category may be applicable in several ways.

One way is automatic acquisition of lexical sets for textbooks authoring. Currently, textbook authors construct lexical sets, and word lists by manual work, relying on their memory and expertise. Language changes dynamically, textbooks have to be reissued and lexical sets needed for them have to be reinvented. Instead, a dynamic method that reflects the modern language use, because it is Web-based, and that takes the typicality of words into consideration will reduce the costs, and will provide richer resources for the text authors' consideration.

Another useful application of the proposed method would be as an extension for a dictionary. It will provide for cases that a certain word belongs to the passive vocabulary, but cannot be retrieved directly. Furthermore, it will be helpful in cases when the word in the target language does not have an equivalent in learner's first language⁴ of the learner, and bilingual dictionary cannot be used for that purpose. For instance, the Russian word for 'light blue' ('голубой' – *goluboy*) is a very basic color name, of similar typicality to such basic colors as 'red' or 'blue'. A possible English equivalent 'azure' exists, but it is much less typical in English. The learner that wants to learn, or reinforce his knowledge about basic colors in Russian will easily retrieve the ubiquitous word for 'light blue' by providing the Russian equivalents for 'blue' and 'red' to WordSets. If the word is already in his passive vocabulary he will recognize it. Otherwise, he will look it up in the bilingual dictionary that will be complementary to WordSets in such case.

Word lists by language teachers provide a good combination of similarity by semantics and by typicality in a way useful for learners, hence being important resources for evaluation. The empirical evaluation provided in this study shows a clear improvement of precision by ranking a set of similar words. It also demonstrates comparability of the established method to GoogleSets and a general conformity with the familiarity

ratings. However, a limited choice of manually constructed word lists as the evaluation data cannot fully reflect its advantages and deficiencies. We plan an extensive evaluation procedure with human subjects that are language learners in the near future.

The scope of the current study is English. However, we believe, that the suggested method may be applied for other languages in a similar manner, given large corpora and a WordNet in another language.

8. References

- [1] S. Banerjee and Ted Pedersen. An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. CILing-02, Mexico, 2002.
- [2] I.A. Bolshakov and S.N. Galicia Haro. How many pages in a given language does Internet have? (In Russian). Computational Linguistics and Intellectual Technologies. Dialogue-2003, pp. 76-82, Nauka, Moscow, Russia, 2003.
- [3] A. Budanitsky and G. Hirst. Evaluating WordNet-based Measures of Lexical Semantic Relatedness. Computational Linguistics, 32(1):13-47, 2006.
- [4] R. Cilibrasi and P. Vitanyi. The ComLearn Toolkit, http://clo.complearn.org/clo/showexpnum/1166446698/experiment_s.html, 2003.
- [5] R. Cilibrasi and P. Vitanyi. The Google Similarity Distance. *IEEE Transactions on Knowledge and Data Engineering* 19(3): 370-383, 2007.
- [6] D. Davidov and Ari Rappoport. Efficient Unsupervised Discovery of Word Categories Using Symmetric Patterns and High Frequency Words. ACL, Sydney, 2006.
- [7] Google™ Sets. <http://labs.google.com/sets>, 2002.
- [8] Jay Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *COLING*, Taiwan, 1997.
- [9] C. Kelly and L. Kelly. <http://www.manythings.org/vocabulary.2005-2006>.
- [10] D. Lin, Automatic Retrieval and Clustering of Similar Words. COLING-ACL, Montreal, 1998.
- [11] D. McCarthy, Rob Koeling, et al. Predominant Word Senses in Untagged Text. ACL. Barcelona, Spain, 2004.
- [12] G.A. Miller et al, WordNet. A Lexical Database for the English Language. Cognitive Science Lab, Princeton University. <http://www.cogsci.princeton.edu/~wn>, 2006
- [13] Paul Nation. Learning Vocabulary in Lexical Sets: Dangers and Guidelines. *TESOL Journal*, v. 9, n. 2, pp. 6-10, 2000.
- [14] Ari Rappoport and V. Sheinman. A Second Language Acquisition Model Using Example Generalization and Concept Categories. Workshop on Psychocomputational Models of Human Language Acquisition, ACL, Ann Arbor, 2005
- [15] J. Weeds and D. Weir. Co-occurrence Retrieval: A Flexible Framework for Lexical Distributional Similarity. *Computational Linguistics*, V. 31, Issue 4, 2005.
- [16] D. Widdows and B. Dorow, A Graph Model for Unsupervised Lexical Acquisition. COLING, Taiwan, 2002.
- [17] Wilson, M.D. The MRC Psycholinguistic Database: Machine Readable Dictionary. *Behavioural Research Methods, Instruments and Computers*, 20(1), 6-11, 1988.
- [18] Y. Yamazaki and D. Mitsuru. Hakase: Basic Japanese for Students. 3A Corporation. Tokyo, 2006.

⁴ By first language we refer to any language that the learner knows, not necessarily one, for this matter