# Unsupervised Learning of General-Specific Noun Relations from the Web

**Gaël Dias[1], Raycho Mukelov[1], Guillaume Cleuziou[2]**
[1] Universidade da Beira Interior, Covilhã, Portugal
{ddg, raicho}@hultig.di.ubi.pt
[2] University of Orléans, Orléans, France
guillaume.cleuziou@univ-orleans.fr

**Abstract**

In this paper, we propose a new methodology based on directed weighted graphs and the TextRank algorithm to automatically induce general-specific noun relations from web corpora frequency counts. Different asymmetric association measures are implemented to build the graphs upon which the TextRank algorithm is applied and produces an ordered list of nouns from the most general to the most specific. Experiments are conducted based on the WordNet noun hierarchy and a quantitative evaluation is proposed.

## Introduction

Taxonomies are crucial for any knowledge-based system. As a consequence, many attempts have been made to automatically produce taxonomies (Grefenstette, 1994), but (Caraballo, 1999) is certainly the first work which proposes a complete overview of the problem by (1) automatically building a hierarchical structure of nouns based on bottom-up clustering methods and (2) labeling the internal nodes of the resulting tree with hypernyms from the nouns clustered underneath by using patterns such as "X is a kind of Y". In this paper, we are interested in dealing with the second problem of the construction of an organized lexical resource i.e. discovering general-specific noun relations, so that correct nouns are chosen to label internal nodes of any hierarchical knowledge base, such as proposed in (Dias et al., 2006).

Most of the works proposed so far have (1) used predefined patterns or (2) automatically learned these patterns to identify hypernym/hyponym relations. From the first paradigm, (Hearst, 1992) first identifies a set of lexico-syntactic patterns that are easily recognizable i.e. occur frequently and across text genre boundaries. These can be called seed patterns. Based on these seeds, she proposes a bootstrapping algorithm to semi-automatically acquire new more specific patterns. Similarly, (Caraballo, 1999) uses predefined patterns such as "X is a kind of Y" or "X, Y, and other Zs" to identify hypernym/hyponym relations. A more challenging task is to automatically learn the relevant patterns for the hypernym/hyponym relations. In the context of pattern extraction, there exist many approaches as summarized in (Stevenson and Greenwood, 2006). The most well-known work in this area is certainly the one proposed by (Snow et al., 2005) who use machine learning techniques to automatically replace hand-built knowledge. By using dependency path features extracted from parse trees, they introduce a general-purpose formalization and generalization of these patterns. (Sang and Hofmann, 2007) use a similar way as (Snow et al., 2006) to derive extraction patterns for hypernym/hyponym relations by using web search engine counts for pairs of words encountered in WordNet. However, the most interesting work is certainly proposed by (Bollegala et al., 2007) who extract patterns in two steps. First, they find lexical relations between synonym pairs based on snippets counts and apply wildcards to generalize the acquired knowledge. Then, they apply a SVM classifier to determine whether a new pair shows a relation of synonymy or not[1].

On the one hand, links between words that result from manual or semi-automatic acquisition of relevant predicative or discursive patterns (Hearst, 1992; Carballo, 1999) are fine and accurate, but the acquisition of these patterns is a tedious task that requires substantial manual work. On the other hand, methodologies to automatically acquire these patterns mostly based on supervised learning (Snow et al., 2005; Snow et al., 2006; Sang and Hofmann, 2007; Bollegala et al., 2007) to leverage manual work still need to build training data. Unlike other approaches, we propose an unsupervised methodology which aims at discovering general-specific noun relations which can be assimilated to hypernym/hyponym relations detection[2]. The advantages of this approach are clear as it can be applied to any language or any domain without any previous knowledge, based on a simple assumption: specific words tend to attract general words with more strength than the opposite. As (Michelbacher et al., 2007) state: "there is a tendency for a strong forward association from a specific term like *adenocarcinoma* to the more general term *cancer*, whereas the association from cancer to *adenocarcinoma* is weak". Based on this assumption, we propose a methodology based on directed weighted graphs and the TextRank algorithm (Mihalcea and Tarau, 2004) to

[1] This technique could easily be extended to hypernym/hyponym relations.

[2] We must admit that other kinds of relations may be covered. For that reason, we will speak about general-specific relations instead of hypernym/hyponym relations.

automatically induce general-specific noun relations from web corpora frequency counts. Indeed, asymmetry in Natural Language Processing can be seen as a possible reason for the degree of generality of terms (Michelbacher et al., 2007). So, different asymmetric association measures are implemented to build the graphs upon which the TextRank algorithm is applied and produces an ordered list of nouns from the most general to the most specific. Experiments have been conducted based on the WordNet noun hierarchy and a quantitative evaluation proposed using the statistical language identification model (Beesley, 1998) as well as a simple list overlapping.

## Asymmetric Association Measures

In (Michelbacher et al., 2007), the authors clearly point at the importance of asymmetry in Natural Language Processing. In particular, we deeply believe that asymmetry is a key factor for discovering the degree of generality of terms. It is cognitively sensible to state that when someone hears about "*mango*", he may induce the properties of a "*fruit*". But, when hearing "*fruit*", more common fruits will be likely to come into mind such as "*apple*" or "*banana*". In this case, there exists an oriented association between "*fruit*" and "*mango*" (mango → fruit) which indicates that "*mango*" attracts more "*fruit*" than "*fruit*" attracts "*mango*". As a consequence, "*fruit*" is more likely to be a more general term than "*mango*". Based on this assumption, asymmetric association measures are necessary to induce these associations. (Pecina and Schlesinger, 2006) and (Tan et al., 2004) propose exhaustive lists of association measures from which we present the asymmetric ones that will be used to measure the degree of attractiveness between two nouns, $x$ and $y$, where $f(.,.)$, $P(.)$ and $P(.,.)$ are respectively the frequency function, the marginal probability function and the joint probability function, and $N$ the total number of pages indexed by the search engine. In our experiments we used $N = 10^{10}$ as a standard commonly used.

$$\text{Braun - Blanquet} = \frac{f(x,y)}{\max(f(x,y)+f(x,\bar{y}), f(x,y)+f(\bar{x},y))} \quad (1)$$

$$\text{J measure} = \max\left[\begin{array}{l} P(x,y)\log\frac{P(y|x)}{P(y)}+P(x,\bar{y})\log\frac{P(\bar{y}|x)}{P(\bar{y})}, \\ P(x,y)\log\frac{P(x|y)}{P(x)}+P(\bar{x},y)\log\frac{P(\bar{x}|y)}{P(\bar{x})} \end{array}\right] \quad (2)$$

$$\text{Confidence} = \max\left[P(x|y), P(y|x)\right] \quad (3)$$

$$\text{Laplace} = \max\left[\frac{N.P(x,y)+1}{N.P(x)+2}, \frac{N.P(x,y)+1}{N.P(y)+2}\right] \quad (4)$$

$$\text{Conviction} = \max\left[\frac{P(x).P(\bar{y})}{P(x,\bar{y})}, \frac{P(\bar{x}).P(y)}{P(\bar{x},y)}\right] \quad (5)$$

$$\text{Certainty Factor} = \max\left[\frac{P(y|x)-P(y)}{1-P(y)}, \frac{P(x|y)-P(x)}{1-P(x)}\right] \quad (6)$$

$$\text{Added Value} = \max\left[P(y|x)-P(y), P(x|y)-P(x)\right] \quad (7)$$

All seven definitions show their asymmetry by evaluating the maximum value between two hypotheses i.e. by evaluating the attraction of $x$ upon $y$ but also the attraction of $y$ upon $x$. As a consequence, the maximum value will decide the direction of the general-specific association i.e. ($x \rightarrow y$) or ($y \rightarrow x$).

## TextRank Algorithm

TextRank is a graph-based ranking algorithm which essentially decides the importance of a vertex within a graph, based on global information recursively drawn from the entire graph. Informally, if $x$ attracts more $y$ than $y$ attracts $x$, we will draw an edge between $x$ and $y$ as follows (x → y) as we want to give more credits to general words. Formally, we can define a directed graph G = ($V$, $E$) with the set of vertices $V$ (in our case, a set of words) and a set of edges $E$ where $E$ is a subset of $V \times V$. In Figure 1, we show the directed graph obtained by using the set of words $V$ = {*isometry*, *rate of growth*, *growth rate*, *rate*} which represents one artificial cluster where "*rate of growth*" and "*growth rate*" are synonyms, "*isometry*" an hyponynym of the previous set and "*rate*" an hypernym of the same set. The weights associated to the edges have been computed by the confidence association measure (Equation 3) based on web search engine counts[3].

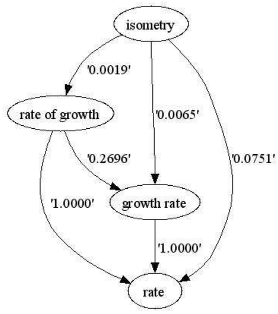


**Figure 1:** Sample Directed Graph.

Figure 1 clearly shows our assumption of generality of terms as the hypernym "*rate*" only has incoming edges whereas the hyponym "*isometry*" only has outgoing edges. Consequently, by applying the TextRank algorithm, we aim at producing an ordered list of words from the most general (with the highest score) to the most specific (with the lowest score). For a given vertex $V_i$ let $In(V_i)$ be the set of vertices that point to it (predecessors), and let $Out(V_i)$ be the set of vertices that vertex $V_i$ points to (successors). The

[3] We used counts for every single word and every pair of words returned by http://www.yahoo.com.

score of a vertex *Vi* is defined in Equation 8 where *d* is a damping factor usually set to 0.85.

$$S(V_i) = (1 - d) + d \times \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} \times S(V_j) \qquad (8)$$

In order to take into account the weights of the edges, a new formula is introduced in Equation 9.

$$WS(V_i) = (1 - d) + d \times \sum_{j \in In(V_i)} \frac{w_{ji}}{\sum_{k \in Out(V_j)} w_{jk}} \times WS(V_j) \qquad (9)$$

After running the algorithm in both cases, a score is computed for each vertex, which represents the "importance" of the vertex within the graph.

| Unweighted | | Weighted | | WordNet | |
|---|---|---|---|---|---|
| $S(V_i)$ | Word | $WS(V_i)$ | Word | Category | Word |
| 0.50 | *Rate* | 0.81 | *rate* | Hypernym | *rate* |
| 0.27 | *growth rate* | 0.44 | *growth rate* | Synset | *growth rate* |
| 0.19 | *rate of growth* | 0.26 | *rate of growth* | Synset | *rate of growth* |
| 0.15 | *isometry* | 0.15 | *isometry* | Hyponym | *isometry* |

**Table 1:** TextRank ordered lists.

As a consequence, after running the TextRank algorithm, in both its configurations, the output is an ordered list of words from the most general one to the most specific one. In table 1, we show both the lists with the weighted and unweighted versions of the TextRank based on the directed graph shown in Figure 1. The results show that asymmetric measures combined with directed graphs and graph-based ranking algorithms such as the TextRank are likely to give a positive answer to our hypothesis about the degree of generality of terms. Moreover, we propose an unsupervised methodology for acquiring general-specific noun relations. However, it is clear that deep evaluation is needed.

## Experiments and Results

Evaluation is classically a difficult task in Natural Language Processing. Human judgment or evaluation metrics are two possibilities. However, human evaluation is time-consuming and generally subjective even when strict guidelines are provided. Thus, in order to validate our assumptions, we propose an automatic evaluation scheme based on statistical language identification techniques (Beesley, 1998) as well as a simple list overlapping.

### Evaluation Measures

To identify the language of a text, a distance between its frequency-ordered list of N-grams and language baseline frequency ordered-lists can be computed. For each N-gram in the test document, there can be a corresponding one in the current language profile it is compared to. N-grams having the same rank in both profiles receive a zero distance. If the respective ranks for an N-gram vary, they are assigned the number of ranks between the two as shown in Figure 2. Finally all individual N-gram rank distances are added up and evaluate the distance between the sample document and the current language profile.

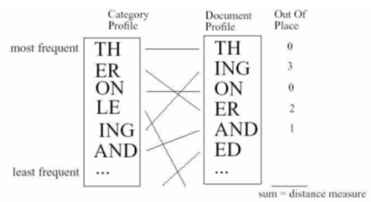


**Figure 2:** Statistical Language Identification

For our purpose, we aim at calculating the distance between the lists of general-specific relations encountered by the TextRank algorithm and the original list given by WordNet. However, we face one problem. WordNet does not give an order of generality inside a synset. In order to avoid this problem, we decided to order the words in each synset by their estimated frequency given by WordNet[4] and their frequency calculated in the web space, as our work is based on document hits. An example of both ordered lists is given in Table 2 showing different results.

| WordNet Estimated Frequency | | Web Estimated Frequency | |
|---|---|---|---|
| Category | Word | Category | Word |
| Hypernym | *statement* | Hypernym | *statement* |
| Synset | *answer* | Synset | *reply* |
| Synset | *reply* | Synset | *response* |
| Synset | *response* | Synset | *answer* |
| Hyponym | *rescript* | Hyponym | *feedback* |
| Hyponym | *feedback* | Hyponym | *rescript* |

**Table 2:** Estimated Frequencies ordered lists.

So, calculating the distance *d(.,.)* between a WordNet ordered list and a list given by our methodology could be done the following way based on Table 3 as shown in Equations 10 and 11.

| Weighted list (A) | WordNet Esti. List (B) | Web Esti. List (C) |
|---|---|---|
| *feedback* | *statement* | *statement* |
| *statement* | *answer* | *reply* |
| *reply* | *reply* | *response* |
| *answer* | *response* | *answer* |
| *response* | *rescript* | *feedback* |
| *rescript* | *feedback* | *rescript* |

**Table 3:** Ordered lists to calculate *d(.,.)*.

$$d(A, B) = 5 + 1 + 0 + 2 + 1 + 1 = 10 \qquad (10)$$

$$d(A, C) = 4 + 1 + 1 + 0 + 2 + 0 = 8 \qquad (11)$$

It is clear that this distance is a penalty factor which must be averaged by the length of the list. For that purpose, we propose the *matching-score(.,.)* in Equation 12 (where *length(.)* is the number of words in a list and *n* is a not null positive integer) which aims at weighting positively the fact that two lists *A* and *B* are similar.

[4] The estimated frequency in WordNet is actually obtained from the SemCor annotated corpus. We use WordNet 2.1.

$$matching-score(A,B) = \begin{cases} 1-\frac{d(A,B)}{2n^2}, length(A)=length(B)=2n \\ 1-\frac{d(A,B)}{2n^2+2n}, length(A)=length(B)=2n+1 \end{cases} \quad (12)$$

We also propose a second evaluation measure which ignores the order of the words and takes into account just the overlapping between two lists (e.g. a given synset and a sub-list derived from the list produced by TextRank). Let's consider we have a hypernym synset consisting of *n* words. We build a second list by taking the first *n* words from the corresponding list generated by TextRank and then compute the *overlapping-score(.,.)* between them. Similarly, we take the last *n* words from the list for the hyponym synset. The aim is to evaluate the ability of the proposed methodology to retrieve true hypernyms and hyponyms by relaxing the order within the lists. Moreover, by computing the *overlapping-score(.,.)* for the seed synset and its corresponding sub-list (which is in fact the rest of the list), we will have a simple indicator showing how well we can partition our list of ranked words. The *overlapping-score(.,.)* measure is defined in Equations 13 and 14 where *A*, *B* and *L* denote lists, *w* a word and |A| the number of words in list A.

$$overlapping-score(A,B) = \frac{\sum_{w \in A} is-in(w,B)}{|A|} \quad (13)$$

$$is-in(w,L) = \begin{cases} 0, w \notin L \\ 1, w \in L \end{cases} \quad (14)$$

## Evaluation Scheme

In order to evaluate our methodology, we randomly extracted 115 seed synsets from which we retrieved their hypernym and hyponym synsets. For each seed synset, we then built the associated directed weighted and unweighted graphs based on the asymmetric association measures referred to in section 2 and ran the TextRank algorithm to produce a general-specific ordered lists of terms. For each produced list, we calculated their *matching-score(.,.)* both with WordNet and Web Estimated Lists for weighted and unweighted graphs. Table 4 presents the average results of the *matching-score(.,.)* for the 115 synsets.

In order to be more precise, we proposed another evaluation scheme by looking at the lists such as a sequence of three sub-lists as presented in Table 5. In fact, we calculated the average *matching-score(.,.)* and the average *overlapping-score(.,.)* for the three sub-lists that are contained in any general-specific list. Indeed, we can look at a list as the combination of the hypernym list, the synset list and the hyponym list. The idea is to identify differences of results in different parts of the lists (e.g. if hypernyms are more easily captured than hyponyms). In Table 5, 6 and 7, we show the results by sub-lists for unweighted and weighted graphs by using respectively the *matching-score(.,.)* and the *overlapping-score(.,.)*.

| Equation | Type of Graph | *matching-score* with WordNet Estimated List | *matching-score* with Web Estimated List |
|---|---|---|---|
| Braun-Blanquet | Unweighted | 51.94 | **52.83** |
| | Weighted | 51.76 | 51.67 |
| J measure | Unweighted | 47.41 | 48.74 |
| | Weighted | 48.32 | 47.81 |
| Confidence | Unweighted | 51.93 | **52.83** |
| | Weighted | 51.76 | 51.67 |
| Laplace | Unweighted | 51.95 | **52.82** |
| | Weighted | 51.95 | **52.82** |
| Conviction | Unweighted | 47.42 | 48.73 |
| | Weighted | 49.38 | 50.06 |
| Certainty Factor | Unweighted | 51.63 | **52.85** |
| | Weighted | 51.29 | 51.16 |
| Added Value | Unweighted | 51.63 | **52.85** |
| | Weighted | 51.20 | 51.57 |

**Table 4.** Average scores in % for entire list comparison.

| Equation | Sub-List | *matching-score* with WordNet Estimated List | *matching-score* with Web Estimated List |
|---|---|---|---|
| Braun-Blanquet | Hypernym | **68.34** | 65.84 |
| | Synset | 55.95 | 54.17 |
| | Hyponym | 56.19 | 54.54 |
| J measure | Hypernym | 61.98 | 60.83 |
| | Synset | 52.47 | 51.12 |
| | Hyponym | 52.91 | 54.62 |
| Confidence | Hypernym | **68.34** | 65.84 |
| | Synset | 55.95 | 54.17 |
| | Hyponym | 56.19 | 54.54 |
| Laplace | Hypernym | **68.34** | 65.84 |
| | Synset | 55.95 | 54.17 |
| | Hyponym | 56.19 | 54.54 |
| Conviction | Hypernym | 62.14 | 60.89 |
| | Synset | 51.75 | 50.62 |
| | Hyponym | 53.87 | 55.68 |
| Certainty Factor | Hypernym | 67.96 | 65.34 |
| | Synset | **56.03** | 54.32 |
| | Hyponym | 56.07 | 54.25 |
| Added Value | Hypernym | 67.32 | 64.70 |
| | Synset | 55.29 | 53.70 |
| | Hyponym | **56.55** | 54.52 |

**Table 5.** Unweighted graphs with matching score.

| Equation | Sub-List | *matching-score* with WordNet Estimated List | *matching-score* with Web Estimated List |
|---|---|---|---|
| Braun-Blanquet | Hypernym | 67.94 | 65.47 |
| | Synset | **56.80** | 54.23 |
| | Hyponym | **56.44** | 54.95 |
| J measure | Hypernym | 64.03 | 61.96 |
| | Synset | 55.96 | 53.12 |
| | Hyponym | 52.83 | 54.08 |
| Confidence | Hypernym | 67.94 | 65.47 |
| | Synset | **56.80** | 54.23 |

| | | | |
|---|---|---|---|
| | Hyponym | **56.44** | 54.95 |
| Laplace | Hypernym | **68.34** | 65.84 |
| | Synset | 55.95 | 54.17 |
| | Hyponym | 56.21 | 54.54 |
| Conviction | Hypernym | 65.72 | 63.57 |
| | Synset | 54.56 | 52.70 |
| | Hyponym | 53.95 | 55.69 |
| Certainty Factor | Hypernym | 67.36 | 64.94 |
| | Synset | **56.80** | 54.51 |
| | Hyponym | 55.93 | 54.37 |
| Added Value | Hypernym | 67.27 | 64.54 |
| | Synset | 56.25 | 54.51 |
| | Hyponym | 56.36 | 54.75 |

**Table 6.** Weighted graphs with matching score.

| Equation | Sub-List | Average *overlapping-score* unweighted | Average *overlapping-score* weighted |
|---|---|---|---|
| Braun-Blanquet | Hypernym | 75.83 | **78.50** |
| | Synset | 54.17 | 55.67 |
| | Hyponym | 55.00 | 54.50 |
| J measure | Hypernym | 70.00 | 71.50 |
| | Synset | 55.00 | **59.00** |
| | Hyponym | 54.33 | **56.00** |
| Confidence | Hypernym | 75.83 | **78.50** |
| | Synset | 54.17 | 55.67 |
| | Hyponym | 55.00 | 54.50 |
| Laplace | Hypernym | 75.83 | 75.83 |
| | Synset | 54.17 | 54.17 |
| | Hyponym | 55.00 | 55.00 |
| Conviction | Hypernym | 70.00 | 73.17 |
| | Synset | 55.00 | 56.83 |
| | Hyponym | 54.33 | 55.17 |
| Certainty Factor | Hypernym | 75.83 | **78.50** |
| | Synset | 55.00 | 55.67 |
| | Hyponym | 54.33 | 54.50 |
| Added Value | Hypernym | 75.83 | **78.50** |
| | Synset | 55.00 | 56.50 |
| | Hyponym | 54.33 | 53.83 |

**Table 7.** Comparison with Overlapping score.

Based on Table 4, the first conclusion to be drawn from our experiments is that unweighted graphs and weighted graphs perform almost the same way in the general case. This clearly shows that the topology of the graph is more important than its weights. However, slight differences can be seen, although they differ from association measure to association measure. Indeed, the biggest difference is 1.33% for the Conviction measure for the case of the Web Estimated List. The second conclusion is the fact that using any of the asymmetric measures does not drastically influence the results. This is a clear consequence of our first conclusion, as the topology is more important than the values given to the edges and most of the asymmetric association measures are able to catch the correct directions of the edges. In fact, the Certainty Factor and the added value, perform best with a maximum *matching-score(.,.)* of 52.85% which means that the list obtained with our methodology overlaps more than a half the Web Estimated List. In fact, we can make two groups of asymmetric association measures although the differences are not so important (the maximum distance between all measures is 5.01%): the best ones are {Braun-Blanquet, Confidence, Laplace, Certainty Factor, Added Value} and the worst results are obtained with {J measure, Conviction}.

An important remark needs to be made at this point of our discussion. There is a large ambiguity introduced in the methodology by just looking at web counts. Indeed, when counting the occurrences of a word like "*answer*", we count all its occurrences for all its meanings and forms. For example, based on WordNet, the word "*answer*" can be a verb with ten meanings and a noun with five meanings. Moreover, words are more frequent than others although they are not so general, unconfirming our original hypothesis. Looking at Table 3, "*feedback*" is a clear example of this statement. As we are not dealing with a single domain within which one can expect to see the "one sense per discourse" paradigm, it is clear that the *matching-score(.,.)* would not be as good as expected as it is clearly biased by "incorrect" counts. For that reason, we proposed to use Web Estimated Lists to evaluate the *matching-score(.,.)*. As expected, the results show improvements although negligible for most measures (the maximum difference is 1.33% for the J measure in the unweighted case). Lately, with (Kilgarriff, 2007), there has been great discussion whether one should use web counts instead of corpus counts to estimate word frequencies. In our study, we clearly see that web counts show evident problems, like the ones mentioned by (Kilgarriff, 2007). However, they cannot be discarded so easily. In particular, we aim at looking at web counts in web directories that would act as specific domains and would reduce the space for ambiguity. Of course, experiments with well-known corpora will also have to be made to understand better this phenomenon. The third conclusion to be drawn from the analysis of the results of Table 5, 6 and 7 is the fact that our methodology is especially tailored to correctly find hypernyms. In particular, we can see that the following association measures {Braun-Blanquet, Confidence, Certainty Factor} give 78.50% overlapping when finding true hypernyms. This result is particularly encouraging reaching high levels of confidence. By taking positions into account, results are not so high but also show high values. In particular, Table 5 shows a maximum *matching-score(.,.)* of 68.34% to discover hypernyms. The fourth conclusion is that the discovery of hyponyms and subsequent list of synonyms (referred until now as synset) is more difficult showing respective maxima of (56.00%) and (59.00%) for the *overlapping-score(.,.)* and (56,55%) and (56,80%) for the *matching-score(.,.)*. Finally, another important remark is that weighted graphs produce better results than unweighted ones unlike what was evidenced by the evaluation of global lists. Moreover, using the WordNet Estimated List also produces better result than the Web Estimated List, unlike what was also shown in Table 4 for the global list evaluation.

## Conclusions and Future Work

In this paper, we proposed a new methodology based on directed weighted/unweighted graphs and the TextRank algorithm to automatically induce general-specific noun relations from web corpora frequency counts. To our knowledge, such an unsupervised experiment has never been attempted so far. In order to evaluate our results, we proposed a new evaluation measure, the *matching-score(.,.)*, based on an adaptation of the statistical language identification model. The results obtained by using seven asymmetric association measures based on web frequency counts showed promising results reaching levels of *matching-score(.,.)* of 68.34% and *overlapping-score(.,.)* of 78.50 % for hypernyms detection. Nevertheless, future work is needed. First, based on the statements of (Kilgarriff, 2007), we aim at reproducing our experiments based on web directories and reference corpora to avoid large scale ambiguity from web counts. Second, the *matching-score(.,.)* generally penalizes the overall results as we still do not have enough consistent way of defining level of generality inside a synset. Finally, we want to deeply study the topologies of the built graphs to understand if simplifications can be made based on their topologies as it is done in (Patil and Brazdil, 2007).

## References

Beesley, K.B. 1998. *Language Identifier: a Computer Program for Automatic Natural-Language Identification of On-line Text*. In Proceedings of the 29th Annual Conference of the American Translators Association.

Bollegala, D., Matsuo, Y. and Ishizuka, M. 2007. *Measuring Semantic Similarity between Words Using Web Search Engines*. In Proceedings of International World Wide Web Conference (WWW 2007).

Caraballo, S.A. 1999. *Automatic Construction of a Hypernym-labeled Noun Hierarchy from Text*. In Proceedings of the Conference of the Association for Computational Linguistics (ACL 1999).

Dias, G., Santos, C., and Cleuziou, G. 2006. *Automatic Knowledge Representation using a Graph-based Algorithm for Language-Independent Lexical Chaining*. In Proceedings of the Workshop on Information Extraction Beyond the Document associated to the Joint Conference of the International Committee of Computational Linguistics and the Association for Computational Linguistics (COLING/ACL 2006), pages. 36-47.

Grefenstette, G. 1994. Explorations in Automatic Thesaurus Discovery. Kluwer Academic Publishers, USA.

Hearst, M.H. 1992. *Automatic Acquisition of Hyponyms from Large Text Corpora*. In Proceedings of the Fourteenth International Conference on Computational Linguistics (COLING 1992), pages 539-545.

Kilgarriff, A. 2007. Googleology is Bad Science. *Computational Linguistics* 33 (1), pages: 147-151.

Michelbacher, L., Evert, S. and Schütze, H. 2007. *Asymmetric Association Measures*. In Proceedings of the Recent Advances in Natural Language Processing.

Mihalcea, R. and Tarau, P. 2004. *TextRank: Bringing Order into Texts*. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), pages 404-411.

Patil, K. and Brazdil, P. 2007. SumGraph: Text Summarization using Centrality in the Pathfinder Network. *International Journal on Computer Science and Information Systems*, 2(1), pages 18-32.

Pecina, P. and Schlesinger, P. 2006. *Combining Association Measures for Collocation Extraction*. In Proceedings of the International Committee of Computational Linguistics and the Association for Computational Linguistics (COLING/ACL 2006).

Sang, E.J.K. and Hofmann, K. 2007. *Automatic Extraction of Dutch Hypernym-Hyponym Pairs*. In Proceedings of Computational Linguistics in the Netherlands Conference (CLIN 2007).

Snow, R., Jurafsky, D. and Ng, A. Y. 2005. *Learning Syntactic Patterns for Automatic Hypernym Discovery*. In Proceedings of the International Committee of Computational Linguistics and the Association for Computational Linguistics (COLING/ACL 2006).

Snow, R., Jurafsky, D. and Ng, A. Y. 2005. *Semantic Taxonomy Induction from Heterogenous Evidence*. In Proceedings of the Neural Information Processing Systems Conference (NIPS 2005).

Stevenson, M., and Greenwood, M. 2006. *Comparing Information Extraction Pattern Models*. In Proceedings of the Workshop on Information Extraction Beyond the Document associated to the Joint Conference of the International Committee of Computational Linguistics and the Association for Computational Linguistics (COLING/ACL 2006), pages. 29-35.

Tan, P.-N., Kumar, V. and Srivastava, J. 2004. Selecting the Right Objective Measure for Association Analysis. Information Systems, 29(4). pages 293-313.