Mapping General-Specific Noun Relationships to WordNet Hypernym/Hyponym Relations

Gaël Dias[1], Raycho Mukelov[1], Guillaume Cleuziou[2]

[1] Centre of Human Language Technology and Bioinformatics, University of Beira Interior
6201-001, Covilhã, Portugal
{ddg, raicho}@hultig.di.ubi.pt
[2] Laboratoire d'Informatique Fondamentale d'Orléans, University of Orléans
45067, Orléans, France
cleuziou@univ-orleans.fr

Abstract. In this paper, we propose a new methodology based on directed 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 both quantitative and qualitative evaluations are proposed.

1 Introduction

Taxonomies are crucial for any knowledge-based system. They are in fact important because they allow to structure information, thus fostering their search and reuse. However, it is well known that any knowledge-based system suffers from the so-called knowledge acquisition bottleneck, i.e. the difficulty to actually model the domain in question. As stated in [3], WordNet has been an important lexical knowledge base, but it is insufficient for domain specific texts. So, many attempts have been made to automatically produce taxonomies [5], but [3] 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 "B is a kind of A".

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 relationships, so that correct nouns are chosen to label internal nodes of any hierarchical knowledge base, such as the one proposed in [4]. Most of the works proposed so far have (1) used predefined patterns or (2) automatically learned these patterns to identify hypernym/hyponym relationships. From the first paradigm, [6] 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, he proposes a bootstrapping algorithm to semi-automatically acquire new more specific patterns. Similarly, [3] uses predefined patterns such as “X is a kind of Y” or “X, Y, and other Zs” to identify hypernym/hyponym relationships. This approach to information extraction is based on a technique called *selective concept extraction* as defined by [11]. Selective concept extraction is a form of text skimming that selectively processes relevant text while effectively ignoring surrounding text that is thought to be irrelevant to the domain.

A more challenging task is to automatically learn the relevant patterns for the hypernym/hyponym relationships. In the context of pattern extraction, there exist many approaches as summarized in [15]. The most well-known work in this area is certainly the one proposed by [13] who use machine learning techniques to automatically replace hand-built knowledge. Using dependency path features extracted from parse trees, they introduce a general-purpose formalization and generalization of these patterns. Given a training set of text containing known hypernym pairs, their algorithm automatically extracts useful dependency paths and applies them to new corpora to identify novel pairs. [12] use a similar way as [14] to derive extraction patterns for hypernym/hyponym relationships by using web search engine counts from pairs of words encountered in WordNet. However, the most interesting work is certainly proposed by [2] who extract patterns in two steps. First, they find lexical relationships 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, based on a feature vector of lexical relationships. This technique could be applied to hypernym/hyponym relationships although the authors do not mention it.

On the one hand, links between words that result from manual or semi-automatic acquisition of relevant predicative or discursive patterns [3], [6] are fine and accurate, but the acquisition of these patterns is a tedious task that requires substantial manual work. On the other hand, works done by [2], [12], [13], [14] have proposed methodologies to automatically acquire these patterns mostly based on supervised learning to leverage manual work. However, training sets still need to be built.

Unlike other approaches, we propose an unsupervised methodology which aims at discovering general-specific noun relationships which can be assimilated to hypernym/hyponym relationships detection[1]. 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 [8] 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”.

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

Based on this assumption, we propose a methodology based on directed graphs and the TextRank algorithm [9] to automatically induce general-specific noun relationships 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 [8]. 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 both quantitative and qualitative evaluations proposed using the statistical language identification model [1].

2 Asymmetric Association Measures

In [8], 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. [10] and [16] 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 $f(.,.)$, $P(.)$, $P(.,.)$ and N are respectively the frequency function, the marginal probability function, the joint probability function, the total of digrams.

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

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

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

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

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

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

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

$$\text{Gini Index} = \max\left[\begin{array}{c} P(x,*)(P(y|x)^2+P(\bar{y}|x)^2)-P(*,y)^2+P(\bar{x},*)(P(y|\bar{x})^2+P(\bar{y}|\bar{x})^2)-P(*,\bar{y})^2, \\ P(*,y)(P(x|y)^2+P(\bar{x}|y)^2)-P(x,*)^2+P(*,\bar{y})(P(x|\bar{y})^2+P(\bar{x}|\bar{y})^2)-P(\bar{x},*)^2 \end{array}\right] \tag{8}$$

$$\text{Collective Strength} = \frac{P(x,y)+P(\bar{x},\bar{y})}{P(x,*)P(y)+P(\bar{x},*)P(*,y)} \times \frac{1-P(x,*)P(*,y)-P(\bar{x},*)P(*,y)}{1-P(x,y)-P(\bar{x},\bar{y})} \tag{9}$$

All nine definitions, except the Collective Strength, 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 upon the direction of the general-specific association i.e. ($x \rightarrow y$) or ($y \rightarrow x$). For the specific case of the Collective Strength both attractions must be evaluated so that the highest value will decide upon the direction of the association.

3 TextRank Algorithm

Graph-based ranking algorithms are essentially a way of deciding the importance of a vertex within a graph, based on global information recursively drawn from the entire graph. Our intuition of using graph-based ranking algorithms is that more general words will be more likely to have incoming associations as they will be associated to many specific words. On the opposite, specific words will have few incoming associations as they will not attract general words (See Figure 1). As a consequence, the voting paradigm of graph-based ranking algorithms should give more strength to general words than specific ones, thus ranking words from general to specific.

For that purpose, we first need to build a directed 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 our case, defined by the asymmetric association measure value between two words). In Figure 1, we show the directed graph obtained by using the set of words *V* = {*isometry*, *rate of growth*, *growth rate*, *rate*} randomly extracted from WordNet 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 evaluated by the confidence association measure (Equation 3) based on web search engine counts[2]. In particular,

[2] We used counts returned by http://www.yahoo.com.

the joint probability between two words, $P(x,y)$, is evaluated by the number of documents retrieved by the Yahoo! search engine for the following query, "*x*" + "*y*", divided by the total number of documents indexed. The same process is applied to evaluate the marginal probabilities P(*x*) and P(*y*).

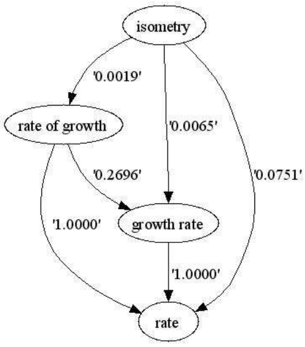

Fig. 1. Directed Graph based on the Confidence measure.

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. Most complicated graphs can be obtained which also confirm our assumption as shown in section 4. As a consequence, by applying a graph-based ranking algorithm, we aim at producing an ordered list of words from the most general (with the highest value) to the most specific (with the lowest value). For that purpose, we present the TextRank algorithm proposed by [9] both for unweighted and weighted directed graphs.

3.1 Unweighted Directed Graph

For a given vertex V_i let $In(V_i)$ be the set of vertices that point to it, and let $Out(V_i)$ be the set of vertices that vertex V_i points to. The score of a vertex V_i is defined in Equation 10 where *d* is a damping factor that can be set between 0 and 1, which has the role of integrating into the model the probability of jumping from a given vertex to another random vertex in the graph[3].

$$S(V_i) = (1 - d) + d \times \sum_{V_j \in In(V_i)} \frac{1}{|Out(V_j)|} \times S(V_j) \tag{10}$$

[3] *d* is usually set to 0.85.

3.2 Weighted Directed Graph

In order to take into account the weights of the edges, a new formula is introduced in Equation 11.

$$WS(V_i) = (1 - d) + d \times \sum_{Vj \in In(V_i)} \frac{w_{ji}}{\sum_{Vk \in Out(V_j)} w_{jk}} \times WS(V_j) \quad (11)$$

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

Table 1. TextRank ordered lists.

Unweighted		Weighted		WordNet	
$S(V_i)$	Word	$WS(V_i)$	Word	Categ.	Word
0.50	*rate*	0.81	*rate*	Hyperonym	*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*

4 Experiments and Results

Evaluation is classically a difficult task in Natural Language Processing. In fact, as human evaluation is time-consuming and generally subjective even when strict guidelines are provided, measures to automatically evaluate experiments must be proposed. In this section, we propose to evaluate the capacity of our approach to map WordNet hypernym/hyponym relations. For that purpose, we introduce two different evaluation schemes.

4.1 Correctness

WordNet can be defined as applying a set of constraints to words. Indeed, if word *w* is the hypernym of word *x*, we may represent this relation by the following constraint $y \succ x$, where › is the order operator stating that *y* is more general than *x*. As a consequence, for each set of three synsets (the hypernym synset, the seed synset and the hyponym synset), a list of constraints can be established i.e. all words of the hypernym synset must be more general than all the words of the seed synset and the hyponym synset, and all the words of the seed synset must be more general than all the words in the hyponym synset. So, if we take the synsets presented in Table 1, we can define the following set of constraints: {*rate* › *growth rate*, *rate* › *rate of growth*, *growth rate* › *isometry*, *rate of growth* › *isometry*}. In order to evaluate our list of words ranked by the level of generality against the WordNet categorization, we just need to measure the proportion of constraints which are respected as shown in Equation (12). We call, *correctness* this measure.

$$correctness = \frac{\#\text{ of common constraint}}{\#\text{ of constraint}} \tag{12}$$

For example, in Table 1, all the constraints are respected for both weighted and unweighted graphs, giving 100% correctness for the ordered lists compared to WordNet categorization.

4.2 Clustering

Another way to evaluate the quality of the ordering of words is to apply hard clustering to the words weighted by their level of generality. By evidencing the quality of the mapping between three hard clusters generated automatically and the hypernym synset, the seed synset and the hyponym synset, we are able to measure the quality of our ranking. As a consequence, we propose to (1) perform 3-means clustering over the list of ranked words, (2) classify the clusters by level of generality and (3) measure the precision, recall and f-measure of each cluster sorted by level of generality with the hypernym synset, the seed synset and the hyponym synset.

For the first task, we use the implementation of the k-means algorithm of the NLTK toolkit[4]. In particular, we bootstrap the k-means by choosing the initial means as follows. For the first mean, we choose the weight (the score) of the first word in the TextRank generated list of words. For the second mean, we take the weight of the middle word in the list and for the third mean, the weight of the last word in the list. For the second task the level of generality of each cluster is evaluated by the average level of generality of words inside the cluster (or said with other words by its mean). For the third task, the most general cluster and the hypernym synset are compared in terms of precision, recall and f-measure as shown in Equation (13), (14) and (15)[5]. The same process is applied to the second most general cluster and the seed synset, and the third cluster and the hyponym synset.

$$precision = \frac{\text{Cluster} \cap \text{Synset}}{|\text{Cluster}|} \tag{13}$$

$$recall = \frac{\text{Cluster} \cap \text{Synset}}{|\text{Synset}|} \tag{14}$$

$$f - measure = \frac{2 \times recall \times precision}{precision + recall} \tag{15}$$

[4] http://nltk.sourceforge.net/

[5] Cluster ∩ Synset means the number of words common to both Synset and Cluster, and |Synset| and |Cluster| respectively measure the number of words in the Synset and the Cluster.

4.2 Quantitative Evaluation

In order to evaluate our methodology, we randomly[6] extracted 800 seed synsets for 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[7] and ran the TextRank.

Results by Constraints

In Table 2, we present the results of the *correctness* for all nine asymmetric measures, both for the unweighted and weighted graphs.

Table 2. Results for the Evaluation by Constraints.

Equation	Type of Graph	Correctness
Braun-Blanquet	Unweighted	65.68%
	Weighted	65.52%
J measure	Unweighted	60.00%
	Weighted	60.34%
Confidence	Unweighted	65.69%
	Weighted	65.40%
Laplace	Unweighted	65.69%
	Weighted	65.69%
Conviction	Unweighted	61.81%
	Weighted	63.39%
Certainty Factor	Unweighted	65.59%
	Weighted	63.76%
Added Value	Unweighted	65.61%
	Weighted	64.90%
Gini Index	Unweighted	65.54%
	Weighted	65.54%
Collective Strength	Unweighted	65.57%
	Weighted	65.57%
Baseline[8]	None	55.68%

Results by Clustering

In Table 3, we present the results of precision, recall and f-measure for both weighted and unweighted graphs for all the nine asymmetric measures. The best precision is obtained for the weighted graph with the Confidence measure evidencing 47.62% and the best recall is also obtained by the Confidence measure also for the weighted graph reaching 47.68%. In particular, the J measure and the Conviction metric perform worst showing worst f-measures.

[6] We guarantee 98% significance level for an error of 0.05 following the normal distribution.
[7] The probability functions are estimated by the Maximum Likelihood Estimation (MLE).
[8] The baseline is the list of words ordered by web hits frequency (without TextRank).

These results also show that the weighting of the graph plays an important issue in our methodology. Indeed, most metrics perform better with weighted graphs in terms of f-measure.

Table 3. Results for the Evaluation by Clustering.

Equation	Graph	Precision	Recall	F-measure
Braun-Blanquet	Unweighted	46.61	46.06	46.33
	Weighted	47.60	47.67	47.64
J measure	Unweighted	40.92	40.86	40.89
	Weighted	42.62	43.71	43.15
Confidence	Unweighted	46.54	46.02	46.28
	Weighted	**47.62**	**47.68**	**47.65**
Laplace	Unweighted	**46.67**	46.11	46.39
	Weighted	46.67	46.11	46.39
Conviction	Unweighted	42.14	41.67	41.90
	Weighted	43.62	43.99	43.80
Certainty Factor	Unweighted	46.48	46.52	46.50
	Weighted	44.84	45.85	45.34
Added Value	Unweighted	46.61	**46.59**	**46.60**
	Weighted	47.12	47.27	47.19
Gini Index	Unweighted	**46.67**	46.11	46.39
	Weighted	46.51	46.02	46.26
Collective Strength	Unweighted	**46.67**	46.11	46.39
	Weighted	46.67	46.11	46.39

In Table 4, 5 and 6, we present the same results as in Table 3 but at different levels of analysis i.e. precision, recall and f-measure at hypernym, seed and hyponym levels. Indeed, it is important to understand how the methodology performs at different levels of generality as we verified that our approach performs better at higher levels of generality.

Indeed, the precision scores go down from 59.50% at the hypernym level to 39.36% at the hyponym level with 46.38% at the seed level. The same phenomenon is inversely true for the recall with 42.93% at the hypernym level, 43.72% at the seed level and 70.80% at the hyponym level. This situation can easily be understood as most of the clusters created by the k-means present the same characteristics i.e. the upper level cluster usually has fewer words than the middle level cluster which in turn has fewer words than the last level cluster. As a consequence, the recall is artificially high for the hyponym level. But on the opposite, the precision is high for higher levels of generality which is promising for the automatic construction of hierarchical thesauri. Indeed, our approach can be computed recursively so that each level of analysis is evaluated as if it was at the hypernym level, thus taking advantage of the good performance of our approach at upper levels of generality[9].

[9] This will be studied as future work.

Table 4. Results at the hypernym level.

Equation	Graph	Precision	Recall	F-measure
Braun-Blanquet	Unweighted	59.38	37.38	45.88
	Weighted	58.75	39.35	47.14
J measure	Unweighted	46.49	37.00	41.20
	Weighted	47.19	41.90	44.38
Confidence	Unweighted	59.20	37.30	45.77
	Weighted	58.71	39.22	47.03
Laplace	Unweighted	**59.50**	37.78	**45.96**
	Weighted	**59.50**	37.78	45.96
Conviction	Unweighted	50.07	35.88	41.80
	Weighted	52.72	40.74	45.96
Certainty Factor	Unweighted	55.90	38.29	45.45
	Weighted	51.64	**42.93**	46.88
Added Value	Unweighted	56.26	**37.90**	45.29
	Weighted	58.21	40.09	**47.48**
Gini Index	Unweighted	**59.50**	37.44	**45.96**
	Weighted	**59.50**	37.44	45.96
Collective Strength	Unweighted	**59.50**	37.44	**45.96**
	Weighted	**59.50**	37.44	45.96

Table 5. Results at the seed level.

Equation	Graph	Precision	Recall	F-measure
Braun-Blanquet	Unweighted	43.05	37.86	40.29
	Weighted	**46.38**	33.14	38.66
J measure	Unweighted	40.82	**43.72**	42.22
	Weighted	43.98	33.89	38.28
Confidence	Unweighted	43.03	37.67	40.17
	Weighted	46.36	33.02	38.57
Laplace	Unweighted	43.10	37.78	40.27
	Weighted	43.10	37.78	40.27
Conviction	Unweighted	40.36	38.02	39.25
	Weighted	42.60	26.39	32.59
Certainty Factor	Unweighted	**44.28**	40.87	**42.51**
	Weighted	44.14	**40.70**	**42.35**
Added Value	Unweighted	44.21	40.74	42.40
	Weighted	45.78	32.90	38.28
Gini Index	Unweighted	43.10	37.79	40.27
	Weighted	42.77	37.25	39.82
Collective strength	Unweighted	43.10	37.78	40.27
	Weighted	43.10	37.78	40.27

Table 6. Results at the hyponym level.

Equation	Graph	Precision	Recall	F-measure
Braun-Blanquet	Unweighted	37.39	62.96	46.92
	Weighted	37.68	70.50	49.12
J measure	Unweighted	35.43	41.87	38.38
	Weighted	36.69	55.33	44.12
Confidence	Unweighted	37.38	63.09	46.95
	Weighted	37.79	**70.80**	**49.27**
Laplace	Unweighted	37.40	**63.11**	46.97
	Weighted	37.40	63.11	46.97
Conviction	Unweighted	35.97	50.94	42.16
	Weighted	35.54	64.85	45.92
Certainty Factor	Unweighted	39.28	60.40	47.60
	Weighted	38.74	53.92	45.09
Added Value	Unweighted	**39.36**	61.15	**47.89**
	Weighted	37.39	68.81	48.45
Gini Index	Unweighted	37.40	**63.11**	46.97
	Weighted	37.25	63.36	46.92
Collective Strength	Unweighted	37.40	**63.11**	46.97
	Weighted	37.40	63.11	46.97

In order to better understand our approach, we present in the next section a qualitative evaluation.

4.3 Qualitative Evaluation

In this section, we intend to illustrate the different situations encountered during our evaluation. We start by showing successful cases. Most of the successful cases were obtained when there are few words to order. In the Example 1 (see also Figure 2), the correct order and clustering was found by our approach i.e. *filter* is the hypernym, *air filter* and *air cleaner* are in the seed synset and *filter tip* is the hyponym. The means are the average levels of generality of the clusters and TextRank shows the values of the ordering of words.

Example 1.

<u>Means</u>: [0.50747799999999, 0.23340649999999, 0.14999999999999]
<u>TextRank</u>: [0.50747799999999, 0.27431299999999, 0.1925, 0.14999999999999]
<u>TextRank sample</u>: ['filter', 'air filter', 'air cleaner', 'filter tip']
<u>Word Clusters</u>: [['filter'], ['air filter', 'air cleaner'], ['filter tip']]
<u>WordNet blueprint synsets</u>: [['filter'], ['air filter', 'air cleaner'], ['filter tip']]

Some other cases were less successful, even when a few words were involved in the evaluation as in Example 2 and Figure 3. In this case, the system successfully categorizes the word *board* but fails to classify *cabinet* and *planning board*. One of the main reasons for this to appear is the fact that *cabinet* is too frequent as it can appear also in French documents and as consequence is incorrectly overestimated. On the other hand, *planning board* is badly classified due to the restriction of the 3-means algorithm. Indeed, in terms of TextRank score it is almost the same as *cabinet* and *advisory board*. But the fact that it is last scored and that the algorithm must choose 3 clusters, artificially misclassifies *planning board*. By looking at the TextRank score, it is even unclear whether *cabinet*, *advisory board* and *planning board* should be separated.

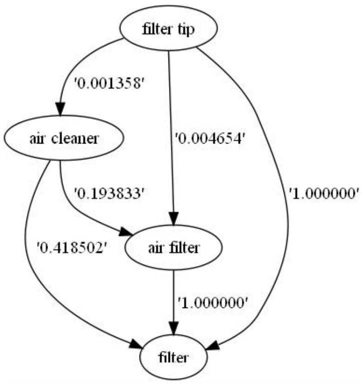

Fig. 2. Directed Graph from Example 1.

Example 2.

<u>Means</u>*: [0.53026700000000, 0.15744200000000, 0.14999999999999]*
<u>TextRank</u>*: [0.530267000000, 0.157872000000, 0.157012000000, 0.149999999999]*
<u>TextRank sample</u>*: ['board', 'advisory board', 'cabinet', 'planning board']*
<u>Word Clusters</u>*: [['board'], ['advisory board', 'cabinet'], ['planning board']]*
<u>WordNet blueprint synsets</u>*: [['board'], ['advisory board', 'planning board'], ['cabinet']]*

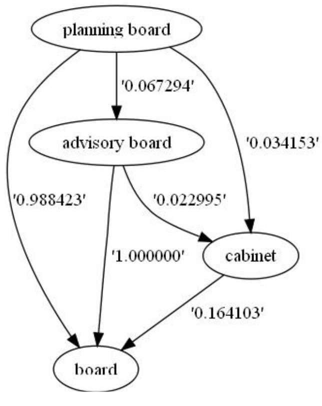

Fig. 3. Directed Graph from Example 2.

In Example 3, we show that in most of the cases, the hypernym cluster is only composed of one word, which in turn is usually correctly classified. Then, the precision of the synset degrades, although it reaches good results if words are not ambiguous like in this example. In Figure 4, we illustrate the corresponding graph.

Example 3.

<u>Means</u>*: [1.115945, 0.45212399999999997, 0.20066549999999997]*
<u>TextRank</u>*: [1.115945, 0.6032140000000, 0.4233080000000, 0.32984999999999998, 0.27204099999999998, 0.232514, 0.20366200000000001, 0.18160899999999999, 0.16416700000000001, 0.14999999999999999]*
<u>TextRank sample</u>*: ['Judaism', 'Jewish religion', 'Orthodox Judaism', 'Hasidim', 'Hasidism', 'Chassidim', 'Hassidim', 'Hebraism', 'Chasidim', 'Jewish Orthodoxy']*
<u>Word Clusters</u>*: [['Judaism'], ['Jewish religion', 'Orthodox Judaism', 'Hasidim'], ['Hasidism', 'Chassidim', 'Hassidim', 'Hebraism', 'Chasidim', 'Jewish Orthodoxy']]*
<u>WordNet blueprint synsets</u>*: [['Judaism', 'Hebraism', 'Jewish religion'], ['Orthodox Judaism', 'Jewish Orthodoxy'], ['Hasidim', 'Hassidim', 'Hasidism', 'Chasidim', 'Chassidim']]*

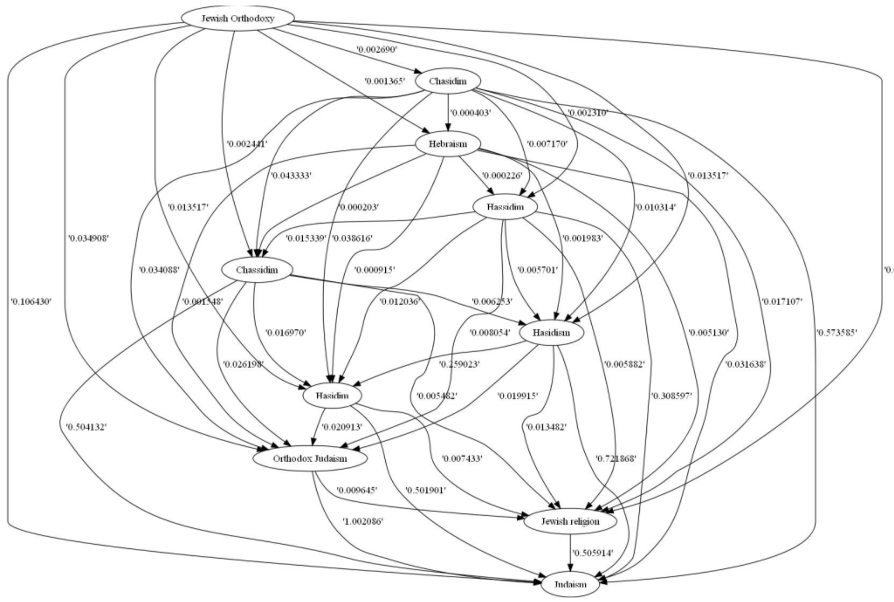

Fig. 4. Directed Graph from Example 3.

In Example 4, we show that when the concepts are at a high level of abstraction, the capability of the approach to classify correctly is weak. In fact, in this case, *instability* is in the hypernym cluster whereas it should be in the hyponym cluster. This shows that *instability* is more frequent than the other words and usually co-occurs with them and not the contrary. In fact, the WordNet classification would be very difficult, even for a human, to be restored.

Example 4.

Means: [0.82191000000000003, 0.37802400000000003, 0.19113825000000001]
TextRank: [0.82191000000000003, 0.444276, 0.31177199999999999, 0.24293999999999999, 0.20036300000000001, 0.17125000000000001, 0.14999999999999999]
TextRank sample: ['instability', 'irresponsibility', 'unreliability', 'undependability', 'irresponsibleness', 'unreliableness', 'undependableness']
Word Clusters: [['instability'], ['irresponsibility', 'unreliability'], ['undependability', 'irresponsibleness', 'unreliableness', 'undependableness']]
WordNet blueprint synsets: [['irresponsibility', 'irresponsibleness'], ['undependability', 'undependableness', 'unreliability', 'unreliableness'], ['instability']]

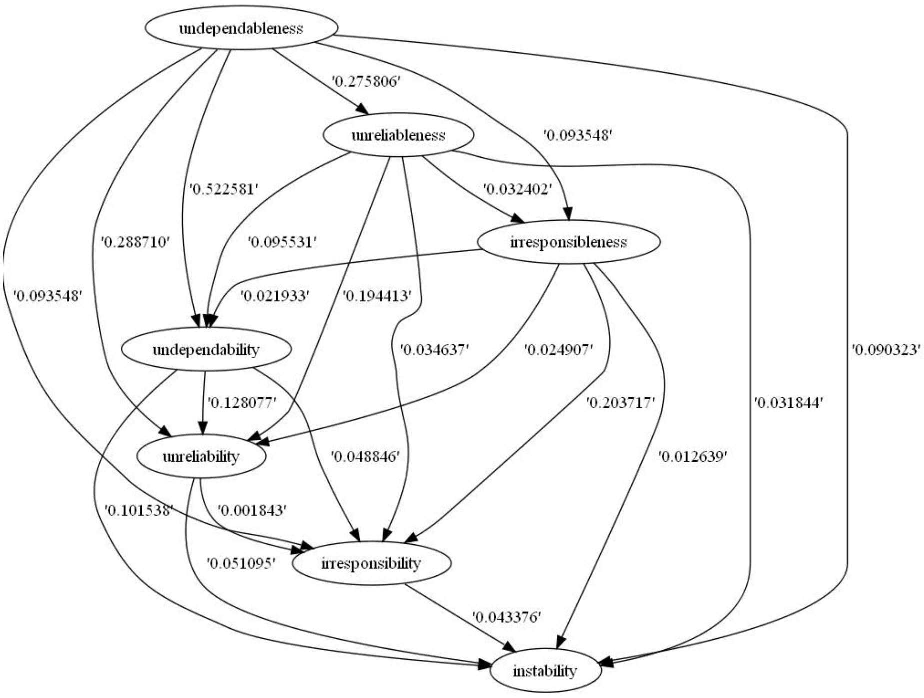

Fig. 5. Directed Graph from Example 4.

4.4 Discussion

An important remark needs to be made at this point of our explanation. 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. 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 results may be biased by "incorrect" counts. One direct implication of this comment is the use of web estimated lists to evaluate the methodology.

Also, there has been a great discussion over the last few months in the corpora list[10] 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 [7]. 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.

Conclusion 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 relationships 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 three different evaluation metrics. The results obtained by using nine asymmetric association measures based on web frequency counts showed promising results reaching levels of (1) constraint coherence of 65.69% and (2) clustering mapping of 59.50% in terms of precision for the hypernym level and 42.72% on average in terms of f-measure.

As future work, we intend to take advantage of the good performance of our approach at the hypernym level to propose a recursive process to improve precision results over all levels of generality.

Finally, it is important to notice that the evaluation by clustering evidences more than a simple evaluation of the word order, but shows how this approach is capable to automatically map clusters to WordNet classification.

References

1. 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, pages 47-54.

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

[10] Finalized by [7].

3. 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).

4. 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), pages. 36-47.

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

6. 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.

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

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

9. 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.

10. 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).

11. Riloff, E. 1993. *Automatically Constructing a Dictionary for Information Extraction Tasks*. In Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI 1993), pages 811-816.

12. 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).

13. 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).

14. 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).

15. 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.

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