An Automatic Method to Identify Antonymy Relations

Cupertino Lucero, David Pinto & Héctor Jiménez-Salazar

Faculty of Computer Science
Benemérita Universidad Autónoma de Puebla,
14 sur y Av. San Claudio. Edif. 135. Ciudad Universitaria,
Puebla, Pue. 72570. México,
Tel. (01222) 229 55 00 ext. 7212 Fax (01222) 229 56 72,
QPrr@hotmail.com, dpinto@cs.buap.mx, hjimenez@fcfm.buap.mx

Abstract. Although WordNet has been used in several applications in many fields of natural language proccessing, the fact that it is not a specialized lexical database (LDB) has lead to the building of LDBs for specific domains. Thus, it is very important to develop automatic methods for recognition of lexical relations embedded in natural language. In this paper we present a new method that identifies pairs of words in opposite relationships from a raw corpus format. The method was applied on pairs of related words obtained from a thesaurus created by Grefenstette's method. We also used some features extracted from word contexts. These features were evaluated by means of the distance between words that appear in the same context, by lexical-syntactic patterns used to match regular expressions with word contexts, and by a lexical co-occurrence network built for each related word. The method was tested on a set of pairs of words taken from an economy corpus, and obtained a 80-percent accuracy, which is highly promising.
Keywords: antonymy relation, lexical co-ocurrence network.

TOPIC: NATURAL LANGUAGE PROCESSING

1 Intoduction

WordNet, the Lexical Data Base (LDB) that covers broad lexical relationships between English words, has had several applications in many fields of natural language processing, such as Information Retrieval, Word Sense Disambiguation, Tuning Lexical Data Bases, and others [4, 8, 9, 11]. However, the fact that there are no specializations of WordNet leads to build LDBs for other domains. In fact, this task needs automatic methods in order to reduce time and effort, and to permit its applicability to others different domains.

The proposals of automatic procedures for identifying underlying lexical relations embedded in natural language closely follow the manual task: the observation of concordances of two terms in order to deduce a relationship between them. For example, P. Hindle [14] proposed an automatic method based on the representation of terms through frequent syntactic features, and thus he determined similarity between nouns. The method developed by G. Grefenstette [1] resembles the previous one. He built thesauri from "poor knowledge", i.e. from a raw corpus of a domain. One of the concepts used in this task involves the idea of mutually near neighbors for a pair of terms: a term that frequently appears in the context of another and reciprocally. In the same way C. Veraschin [5] enriched the related word pairs related with the Brazilian language, based on Grefenstette's method and enriching the features with prepositional phrases. Perhaps the most known method is that proposed by M. Hearst [7] which uses lexical syntactic patterns that match contexts of words in order to identify hyponymy relations. Also, Sanderson and Croft [16] discovered hyponyms based on the notion of subsumption: x subsumes y if texts that contain y imply a high probability as that of the same texts that contain x. More recently, in [6], the subsumption notion was combined with clustering techniques to determine hyponymy and synonymy relations of word pairs in their different word senses. An approach that decides if a pair of words are antonyms, uses conceptual vectors [12], decomposing each term of the pair through an MRD and a thesaurus. This method determines if the pair holds the antonymy relation using operations in a vectorial space as cosine.

In this paper, method that identifies antonyms is presented (we also considered complementaries, as "true" and "false" [13]). This method is based on the representation of terms, previously recognised as related although their relation is unknown, by means of three features extracted from contexts of word pairs: Inversely Proportional Distance between pair words, some Lexical Syntactic Patterns, and significance determined by a Lexical Co-ocurrence Network built for each word. The corpus from which we extracted related pairs also provides the context for determining features. We took a set of pairs to be used as positive examples. From positive examples, the necessary parameters were defined in order to build thresholds useful in the classification.

This problem is difficult to solve because antonyms and synonyms present similar features. Actually, Lexical Co-Ocurrence Networks were used by Philip Edmonds [3] to choose the most adequate synonym in a context. Thus, it is

necessary to discern between these two types of relationships. Specifically, D. Cruse [13] comments on oppositives:

> "... in respect to all other features, they are identical, hence their semantic closeness; along the dimension of difference, they occupy opposing poles, hence the feeling of difference."

For some types of antonyms, the above mentioned may be expressed as L. Wanner [10] did: for richer antonyms x and y their features hold $x = ABC$ and $y = AB\neg C$.

The features used in the classification task are described in section 2 of this paper. Section 3 indicates how to calculate weights associated to features, and a test of the classification procedure on a sample of related pairs is shown. At the end, we present our conclusions.

2 Features Used to Identify Antonyms

Our classification method is based on a score function composed of features weights. The method uses thresholds determined by positive examples. We will describe each one of these features needed in the computation of the score aimed at classifying a pair of words.

Inversely Proportional Distance (IPD). This feature is inspired by the observation of contexts that contain related words. IPD represents how close two words are. In contexts of related words, words in antonymy relationships very frequently occurr at a very close distance. Later observation is supported by the use of antonyms with purposes such as contrastiveness. We define IPD in the following manner: the distance between the related words (number of words that separate them) is complemented with respect to the maximum distance given by positive examples. We took the maximum value of complementary distances in positive examples as:

$$\Delta_M = \max_{(x_1,y_1)\in Pos} \{ \max_{(x_2,y_2)\in Pos} \bar{\Delta}(x_2, y_2) - \bar{\Delta}(x_1, y_1)\} \quad (1)$$

where $\bar{\Delta}(x, y)$ is the average distance between words x and y in their contexts.

Lexical Syntactic Patterns (LSP). LSPs have been introduced in P. Hearst's work [7] with good results in the identification of hyponyms. We identified several patterns antonym context guided by cue terms, punctuation, and distance between related words. LSPs are represented as regular expressions. A sample of these patterns is given in Table 1, which shows the following cue words as part of regular expressions: *pero* 'but', *desde* 'from', *hasta* 'to', *sino* 'but rather', *y* 'and', *o* 'or'.

Lexical Co-ocurrence Network (LCN). Due to the fact that antonyms present similar behavior as synonyms, we had to discern between these two relationships. Therfore, we decided to manage a representation for words that might

Nr	Regular Expression	Weight
1	Ant1 word*, **pero** word* Ant2	5
2	**desde** word* Ant1 **hasta** word* Ant2	4
3	Ant1 word* [,|;] **sino** word* Ant2	5
4	Ant1 word{0,4}[**y**|**o**] word{0,4} Ant2	1

Table 1. Regular expressions and its weights.

contain information used in synonymy problems. LCN was used in our application since it was useful in the solution for choosing the most typical synonym in a context [3]. The procedure used to calculate an LCN for a term, named *root*, is as follows:

1. We considered the sentences for forming the context of the root. The sentences are taken from the corpus (used in our thesaurus construction) $\mathcal{C}$ that contain the root:

$$A_1(x) = \{y | x \text{ and } y \text{ co-occur in a sentence of } \mathcal{C}\} \tag{2}$$

2. The context of the root is cleaned, discarding all words whose mutual information [2] is lower than 3; namely first-order association words, for example:

$$A'_1(x) = \{y | y \in A_1(x) \wedge MI(x, y) > 5\} \tag{3}$$

3. The process is repeated for words $y \in A'_1(x)$ (second-order association words). This depends on the desired level of the LCN. In general, $n-$order association words for x are determined according to:

$$A'_n(x) = \bigcup_{y \in A'_{n-1}(x)} A'_1(y) \tag{4}$$

Because of the necessity to discern the synonymy or antonymy relationships between words (say a_1 and a_2), the LCN of words was used to decide wheter they may be considered as synonyms. One way of doing this is calculating the relative significance between words. Significance, in terms of LCN, is defined in [3] with the purpose of calculating the significance of a word w in a context X, which is based on the significance between words w and x, where x is supposed to be in the LCN of w. Let us suppose that each arc of LCN is weighted (this topic will be explained in Section 3), and let $P = (w_0, w_1, \ldots, w_n)$ be the minimum cost path from $w_0 = w$ to $w_n = x$. The significance of w and x is:

$$sig(w, x) = \frac{1}{d^3} \sum_{w_i \in P} \frac{t(w_{i-1}, w_i)}{i}, \tag{5}$$

where $t(w_{i-1}, w_i)$ is the $t-$score defined in [2] (see appendix for details).

Given words a_1 and a_2 with LCN nodes $L(a_1)$ and $L(a_2)$, respectively, we expect them to have a high significance if the sum of word significance in $L(a_1) \cap L(a_2)$ for both LCNs is high. Therefore, it is necessary to know when the

significance is high. We will refer to the total significance in order to calculate relative significance. Total significance, s_t, is computed by adding all weights of both $L(a_1)$ and $L(a_2)$. In summary, relative significance between words a_1 and a_2 is defined as:

$$s_r(a_1, a_2) = \frac{1}{s_t} \sum_{w \in \{a_1, a_2\}, x \in L(a_1) \cap L(a_2)} sig(w, x), \tag{6}$$

We can then say that features help to determine the global score of a related pair in order to know wheter they are antonyms.

3 Determination of Global Score

Global score takes into account each feature described above. According to values observed in positive examples, a weight is assigned to each feature. Highest weights express that a said feature value was observed in positive examples. For example, a low weight given to a regular expression indicates that some negative examples match such a pattern.

Global score $S_g(a_1, a_2)$ must group all feature values into only one value. This score is calculated by adding the feature weights:

$$S_g(a_1, a_2) = W_{er}(a_1, a_2) + W_d(a_1, a_2) + W_{net}(a_1, a_2), \tag{7}$$

where $W_{er}(a_1, a_2)$ is the weight obtained by: the regular expressions that match contexts wich contain both a_1 and a_2, the weight given by the inversely proportional distance ($W_d(a_1, a_2)$), and an inversely proportional value to $s_r(a_1, a_2)$ ($W_{net}(a_1, a_2)$). Each weight is normalized within the range $[0, 1]$, so S_g is less than 3. Let us describe each one of these values.

Given E, the set of all regular expressions that match the contexts of a_1 and a_2; $weight(e)$, the weight of e (some examples are shown in Table 1); and $fr(e)$ the relative frequency that matches e with contexts, we defined $W_{er}(a_1, a_2) = \sum_{e \in E} weight(e) \cdot fr(e)$.

$W_d(a_1, a_2)$ considers maximum scores for positive examples.

Thus, $W_d(a_1, a_2)$ comes from Δ_M in order to attain a normalized value in $[0, 1]$:

$$W_d(a_1, a_2) = \frac{\Delta_M - \bar{\Delta}(a_1, a_2)}{\Delta_M} \tag{8}$$

And $W_{net}(a_1, a_2)$ follows a similar calculation:

$$W_{net}(a_1, a_2) = \frac{\max\limits_{(x,y) \in Pos} \{s_r(x, y)\} - s_r(a_1, a_2)}{\max\limits_{(x,y) \in Pos} \{s_r(x, y)\}} \tag{9}$$

In the test, we used a corpus composed by 26297 sentences, 11675 terms (including proper nouns, after removing stopwords and lematizing the rest). The training set is composed of 15 pairs, 10 positives and 5 negatives, wich was then

used to tune thresholds. Our test set was composed by 8 pairs of antonyms and 10 pairs of non-antonyms. Table 3 shows some examples of the method application. It is important to note that only four pairs was incorrectly classified, and that the method also rejected hyponyms.

Pair Words	Distance	LSP	LCN	Veredict
Absoluto-Relativo	0.64814815	0.79333333	0.97665313	well assigned
Compra-Venta	0.59259259	0.16666667	0.47302784	well assigned
Natural-Artificial	0.68518519	0.58	0.9724478	well assigned
Consumidor-Productor	0.62962963	0.56	0.47911833	well assigned
Escasez-Abundancia	1	1	0.36180394	well assigned
General-Particular	0	0.50666667	1	well assigned
Máximo-Mínimo	0.51851852	0.26666667	0.64022622	well assigned
Positivo-Negativo	0.44444444	0.80666667	0	well assigned
Verdad-Mentira	0.72222222	0.66666667	0.9650522	well assigned
Vida-Muerte	0.88888889	0.86666667	0.77392691	well assigned
Corrección-Ajuste	0.75925926	0.2	0.08164153	well assigned
Hombre-Humano	-0.38888889	0.10666667	0.97056265	well assigned
Mercancía-Producto	-0.05555556	0.18666667	0.83236659	well assigned
Moneda-Dinero	0.05555556	0.02	0.87558005	well assigned
Obrero-Trabajador	-0.48148148	0	0.86934455	well assigned

Table 2. Training Set: first 10 pairs are antonyms.

4 Conclusions

We have presented a method that identifies terms in relationships of opposite-ness. The solution to the problem of antonym identification is very important, since it interacts with other methods and might therefore reinforce or debilitate the hypothesis about words relationships. The classification method begins with a pair of words previously recognized as related. It makes use of features extracted from the contexts of related words: Inversely Proportional Distance between words in their context, Lexical Syntactical Patterns that match with contexts, and Lexical Co-Ocurrence Networks built on each word in a pair. The classification method parameters were provided by a set of positive examples. Currently, we have tested the method with a sample of pairs taken from an economy corpus, and obtained a highly promising accuracy (80-percent).

Although our results are interesting, it is necessary to give them more generality. The experiment was accomplished in a single corpus: an economy domain. We suppose that antonyms where used with some degree of homogeneity in that corpus. However it would be necessary to use prototypical examples of usual language. For example, glosses of WordNet may provide positive examples for the selection from a large corpus of antonyms contexts. Therefore, one would be able to tune antonyms for specialized domains.

Pair Words	Distance	LSP	LCN	Veredict
Bajo-Alto	0.53703704	0.2	0.62601508	well assigned
Activo-Pasivo	0.67037037	0.76666667	0.28741299	well assigned
Grande-Pequeño	0.47407407	0.53333333	0.97317285	well assigned
Oferta-Demanda	0.79259259	0.76666667	0.04350348	well assigned
Pregunta-Respuesta	0.53703704	0.02666667	0.1712587	wrong
Público-Privado	0.48148148	0.53333333	0.75710557	well assigned
Social-Individual	0.27407407	0.63333333	1.0549594	well assigned
Interior-Exterior	0.50740741	0.73333333	0.62427494	well assigned
Confianza-Fe	-2.05555556	0	0.90762761	well assigned
Crédito-Préstamo	0.33333333	0.33333333	0.36107889	well assigned
Cultura-Democracia	0.7037037	0.26666667	0.67444896	wrong
Harina-Trigo	0.6	0.43333333	0.05191415	well assigned
Inversión-Gasto	0.53148148	0.1	0.33990719	well assigned
Miembro-Comunidad	0.90740741	0	0.70562645	well assigned
Pobreza-Problema	0.56851852	0.03333333	1.03045244	wrong
Productor-Benefactor	0.32222222	0.16666667	0.34353248	well assigned
Rasgo-Característica	0.85185185	0	0.72186775	wrong
Semana-Día	0.30185185	0	0.50101508	well assigned

Table 3. Test Set: first 8 pairs are antonyms.

References

1. G. Grefenstette: *Explorations in Automatic Thesaurus Discovery*, Kluwer Academic Publishers, Boston Hardbound, ISBN 0-7923-9468-2 July 1994.
2. Church, Kenneth Ward; Gale, William; Hanks, Patrick; Hindle, Donald; Moon, Rosamund: "Lexical Substitutability, In: Atkins, B. T. S.; Zampolli, Antonio (eds.): *Computational Approaches to the Lexicon*. Oxford University Press, pp. 153-180, 1994.
3. Edmonds P.: "Choosing the word most typical in context using a lexical cooccurrence network, *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, Madrid, pp. 507-50, 1997.
4. Morato J., Marzal M.A., Lloréns J., Moreiro J.: "WordNet Applications, Petr Sojka, Karel Pala, Pavel Smrc, Christiane Fellbaum, Piek Vossen (Eds.): *Proceedings GWC 2004*, pp. 270-278, 2004.
5. Caroline Varaschin Gasperin Vera Lcia Strube de Lima, "Experiments on Extracting Semantic Relations from Syntactic Relations, *CiCLing 2003*, LNCS 2588, P. 314-324, 2003.
6. Jiménez-Salazar, H., "A Method of Automatic Detection of Lexical Relationships Using a Raw Corpus, *CiCLing 2003*, LNCS 2588, P. 325-328, 2003.
7. Hearst, M.: "Automatic acquisition of hyponyms from large text corpora. *Proceedings of the 14th International Conference on Computational Linguistics*, Nantes, France, 1992.
8. Yorick Wilks, Roberta Catizone: "Lexical Tuning. *CICLing 2002*, 106-125, 2002.
9. Paolo Rosso, Francesco Masulli, Davide Buscaldi, Ferran Pla, Antonio Molina: "Automatic Noun Sense Disambiguation, *CICLing 2003*: 273-276
10. L. Wanner: *Lexical Functions in Lexicography and Natural Language Processing*, John Benjamins Publishing Company, 1996.

11. Hearst, M.: "Automated Discovery of WordNet Relations", in *WordNet and Electronic Lexical Database*, C. Fellbaum (Ed.), The MIT Press, 1999, P. 131-152.
12. Schwab, D.,Lafourcade, M., Prince, V.: "Antonymy and Conceptual Vectors", in *the Proceedings of the 19th Conference on Computational Linguistics*, 2002, P. 904-910.
13. Cruse, D.: *Lexical Semantics*, Cambridge , Cambridge University Press, 1986.
14. Hindle, D.: "Noun Classification from Predicate-Argument Structures", in *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, 1990, P. 268-275.
15. Hearst, M.: "Automatic Acquisition of Hyponyms from Large Text Corpora", in *the Proceedings of the Fourteenth International Conference on Computational Linguistics*, , , 1992, P. .
16. Sanderson M., Croft B.: "Deriving concept hierarchies from text, In *Proceedings of the 22 "a Annual lnt. ACM S1G1R Conference on Research and Development in Information Retrieval*, pp. 206 - 213, Berkeley, CA, August 1999.

Appendix

Once we build an LCN, the score of each arc t is calculated by giving two terms in association ($w_{i-1} \in A'_{i-1}(x)$ and $w_i \in A'_i(x)$) on levels $i-1$ and i of the network with root x. Their score ($t(w_{i-1}, w_i)$), according to [2], is given as follows:

$$t(w_{i-1}, w_i) = \frac{P(w_{i-1}, w_i) - P(w_{i-1}) \cdot P(w_i)}{\sqrt{(\sigma^2(P(w_{i-1}, w_i)) + \sigma^2(P(w_{i-1}) \cdot P(w_i)))}} \tag{10}$$

with $P(w_{i-1}, w_i) = \frac{fr(w_{i-1}, w_i)}{N}$ and $\sigma^2 P(w_{i-1}, w_i) \cong N \cdot P(w_{i-1}, w_i)$.

The Mutual Information formula used in this work is defined in [2]:

$$MI(x, y) = log_2 \left(\frac{N \cdot fr(x, y)}{fr(x) \cdot fr(y)} + 1 \right) \tag{11}$$

N is the total number of sentences contained in the corpus, $fr(x, y)$ is the number of sentences that contain both words x y y, and $fr(x)$ y $fr(y)$ is the number of sentences that contain the term x y y respectively.

In Figures 1 and 2 we can see fragments of LCNs for the words *costo* 'cost' and *precio* 'price', respectively.

In figure 1, it is particularly possible to see that the nodes that belong to the same level of *beneficio* 'benefit' are first-order association terms with respect to the root, and the significance between root term and each of its first-order terms is labeled on every arc. In this case, the significance and score of t are equal.

Dotted arrows in Figures 1 and 2 illustrate the order level for some words. On the other hand, dotted lines indicate a relationship without value between two nodes (derived from a significance value less than the value obtained in the other path). In figure 1 for example, the significance value for *insumo* and *recibir* is 0.24, whereas the significance value for *beneficio* 'benefit' and *recibir* 'receive' is 0.59, which is greater than the first value, and therefore the path between *insumo* and *recibir* 'receive' is ignored.

The pair (x, y) that labels the path between two nodes of the LCN represents the value x given by the score t (eq. 10) with respect to the linked nodes, and the significance value y between the root term and each node (obtained from the formula given in (eq 10)). The pair that labels the link between *salario* 'salary' and *marginal* 'marginal' in Figure 1 indicates a score of $t = 1.23$, and a significance value of $s = 0.244$ between the root term *costo* 'cost' and the word *marginal* 'marginal'.

For instance, the significance value between the network root *costo* 'costo' and the node *fijo* 'fixed', is determined as follows:

$Sig(costo, fijo) = \frac{1}{3^3}(t(costo, beneficio) + t(beneficio, salario)/2+$
$+t(salario, fijo)/3) \approx \frac{1}{27}(4.57 + 1.63 + 0.44) \approx 0.246$

After determining all the significance values of both co-ocurrence networks, the maximum possible score of significance was calculated, by adding all the values of both networks. In order to determine the compatibility percentage between both related words, the sum of significance value of all the words obtained by an intersection between both networks is calculated and this result is then compared with the maximum possible score of significance in order to obtain its percentage.

A hypothesis of this work is that two terms in synonymy relationship must have a high number of common words in their LCN. Furthermore, if these co-ocurrence networks would have identical words and weights, then both root terms would be perfect synonyms. We would expect that something different happens for two terms with a relationship other than that of synonymy.

The fragments of LCN shown in Figures 1 and 2 have a maximum posible score of 2.548 and a compatibility percentage of 42.2%. The complete network for the words *costo* 'cost' and *precio* 'price' have 1290 and 980 nodes, respectively.

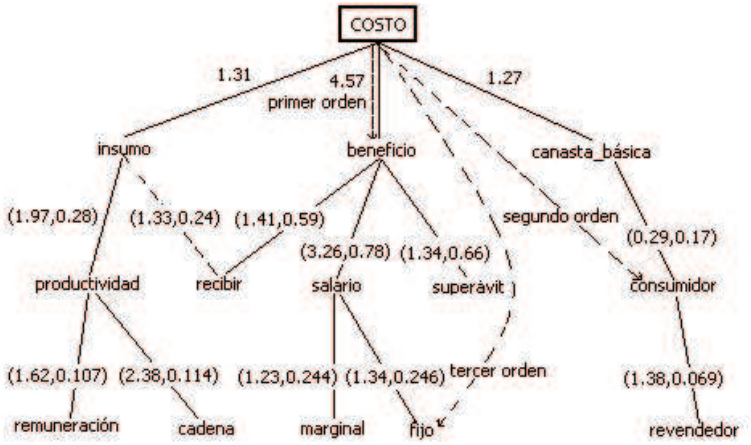

Fig. 1. Fragment of the LCN for the word *costo* 'cost'

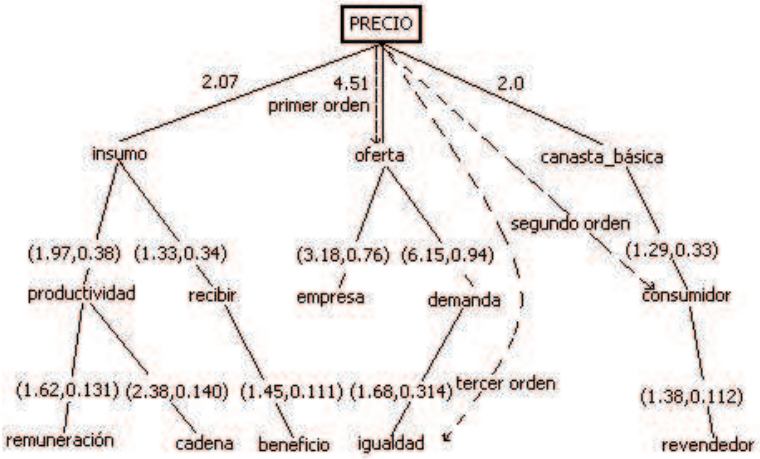

Fig. 2. Fragment of the LCN for the word *precio* 'price'