Induction of Semantic Classes from Natural Language Text

Dekang Lin and Patrick Pantel
University of Alberta
Department of Computing Science
Edmonton, Alberta T6H 2E1 Canada
{lindek, ppantel}@cs.ualberta.ca

ABSTRACT

Many applications dealing with textual information require classification of words into semantic classes (or concepts). However, manually constructing semantic classes is a tedious task. In this paper, we present an algorithm, UNICON, for UNsupervised Induction of CONcepts. Some advantages of UNICON over previous approaches include the ability to classify words with low frequency counts, the ability to cluster a large number of elements in a high-dimensional space, and the ability to classify previously unknown words into existing clusters. Furthermore, since the algorithm is unsupervised, a set of concepts may be constructed for any corpus.

1. INTRODUCTION

Many applications dealing with textual information require classification of words into semantic classes (or concepts). Text mining systems often convert text into a set of features, many of which are defined in terms of semantic classes. In information extraction and question answering, many of the pattern matching rules make use of semantic classes such as management positions, expenditures, art work, etc. [7][20].

Manually constructing semantic classes is a tedious task. Most attempts to automatically construct semantic classes have relied on the Distributional Hypothesis [8] that words that appear in similar contexts are semantically similar. Typically, these algorithms output a similarity matrix that can be used to retrieve the most similar words of a given word as well as the similarity values. There are several drawbacks of such similarity matrices.

Firstly, a threshold is required so that all similarity values lower than the threshold are considered to be 0. Since the threshold is uniformly applied to all the words, it is impossible to use it to separate good similar words from bad ones. For example, the most similar words of Beethoven, obtained from [13], are (the number following each word is its similarity to Beethoven):

```
{Brahms 0.2, Mozart 0.189, Mahler 0.168, Bach
0.146, Schubert 0.142, Prokofiev 0.138,
Tchaikovsky 0.137, Wagner 0.092, chamber
music 0.091, Concerto 0.076, cello 0.074,
Shakespeare 0.074, sonata 0.068, Shakespeare
0.063, Napoleon 0.043, "tough guy" 0.042,
grandparent 0.041 …}
```

A human editor will probably select the words up to Wagner as the similar words. However, if the threshold is set to 0.091, it will fail on other lists of similar words.

Secondly, since similar words are often similar in different senses, a list of similar words does not easily represent concepts [18]. In applications such as information extraction and question answering, concept grouping is critical.

Finally, the distributional word similarity algorithms tend to break down on infrequent words. Such words have few features. Often, words that accidentally share these features will be considered its similar words.

Clustering a large number of elements in a high-dimensional space presents a serious challenge. For example, *CLIQUE* [1] is a clustering algorithm specifically designed to handle high-dimensional spaces. The experiments in [1] involved 10,000 elements in a 100-dimension space. In one of our experiments, we are faced with 146,290 elements and 1,639,996 dimensions. In a two-step clustering algorithm, McCallum et al. [15] first use a computationally inexpensive similarity measure to create a set of overlapping *canopies*. A canopy is a subset of elements that are similar to a central element. The assumption is that two elements not belonging to the same canopy do not belong in the same cluster. Using a traditional clustering algorithm (e.g. Greedy Agglomerative Clustering or K-means) each canopy may then be separately clustered.

In this paper, we present an unsupervised algorithm, UNICON, which overcomes the former limitations. It can be used for inducing a set of concepts, each consisting of a cluster of words. We iteratively apply a clustering algorithm, called CLIMAX, merging clusters by computing and comparing their centroids. Since the algorithm is unsupervised, a set of concepts may be constructed for any corpus.

The remainder of this paper is organized as follows. In the next section, we review previous work in automatic thesaurus construction. Section 3 describes the collocation database required by our system and in Section 4, we present our algorithms. The evaluation of our concept induction algorithm is shown in Section 5. Finally, we conclude with a discussion of future work.

KDD 01 San Francisco CA USA
Copyright ACM 2001 1-58113-391-x/01/08…$5.00

Table 1. Excerpts of entries in the collocation database for *duty* and *responsibility* [12].

DUTY		RESPONSIBILITY	
modified-by adjectives	fiduciary 319, active 251, other 82, official 76, additional 47, administrative 44, military 44, constitutional 41, reserve 24, high 23, moral 21, double 16, day-to-day 15, normal 15, specific 15, assigned 14, extra 13, operating 13, temporary 13, corporate 12, peacekeeping 12, possible 12, regular 12, retaliatory 12, heavy 11, routine 11, sacred 11, stiff 11, congressional 10, fundamental 10, hazardous 10, main 10, patriotic 10, punitive 10, special 10, …	modified-by adjectives	more 107, full 92, fiduciary 89, primary 88, personal 79, great 69, financial 64, fiscal 59, social 59, moral 48, additional 46, ultimate 39, day-to-day 37, special 37, individual 36, legal 35, other 35, corporate 30, direct 30, constitutional 29, given 29, overall 29, added 28, sole 25, operating 23, broad 22, political 22, heavy 20, main 18, shared 18, professional 17, current 15, federal 14, joint 14, enormous 13, executive 13, operational 13, similar 13, administrative 10, fundamental 10, specific 10, …
object-of verbs	have 253, assume 190, perform 153, do 131, impose 118, breach 112, carry out 79, violate 54, return to 50, fulfill 44, handle 42, resume 41, take over 35, pay 26, see 26, avoid 19, neglect 18, shirk 18, include 17, share 17, discharge 16, double 16, relinquish 16, slap 16, divide 14, split 13, take up 13, continue 11, levy 11, owe 10, …	object-of verbs	have 747, claim 741, take 643, assume 390, accept 220, bear 187, share 103, deny 86, fulfill 53, meet 48, feel 47, retain 47, shift 47, carry out 45, take over 41, shoulder 29, escape 28, transfer 28, delegate 26, give 25, admit 23, do 21, acknowledge 20, exercise 20, shirk 20, divide 19, get 19, include 19, assign 18, avoid 17, put 17, recognize 17, hold 16, understand 16, evade 15, disclaim 12, handle 12, turn over 12, become 11, expand 11, relinquish 11, show 11, violate 11, discharge 10, duck 10, increase 10, …

2. Previous Work

There have been several approaches to automatic thesaurus construction, mostly for information retrieval. Many algorithms are based on Harris' Distributional Hypothesis [8] and rely on the similarity between terms by constructing a similarity matrix.

Salton et al. [19] developed a term dependence model based on relevance judgements targeted for information retrieval systems. Term probabilities are estimated using their frequencies in relevant and non-relevant documents. This model was later extended to use term discrimination values to compute the similarity matrix and cluster terms [5]. Low-frequency terms in clusters were then used to generate the thesaurus classes. These methods are unsuitable for our problem since relevance judgements are unavailable.

Bayesian networks have also been used to discover patterns in term usage. Park [17] modelled the similarity distribution among terms using a Bayesian network built from local term dependencies. Compared to previous approaches, this system had the advantage of handling low-frequency terms.

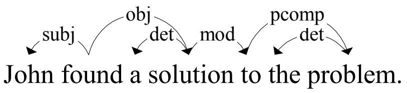
Jing and Croft [11] proposed a thesaurus construction algorithm using co-occurrence frequencies (lexical associations) and text feature recognition such as terms and parts of speech. Using only syntactic information, Grefenstette [6] used a weighted Jaccard measure and Lin [13] proposed an information-theoretic similarity measure to compute the similarity matrix. Chen et al. [4] proposed a three-step algorithm that performs automatic indexing and cluster analysis.

3. Resources

The input to our algorithms includes a collocation database and a similarity matrix, both available on the Internet[1].

A dependency relationship [9][10][16] is an asymmetric binary relationship between a word called **head**, and another word called **modifier**. The structure of a sentence can be represented by a set of dependency relationships that form a tree. A word in the sentence may have several modifiers, but each word may modify at most one word. The root of the dependency tree does not modify any word. It is also called the head of the sentence.

For example, the following diagram shows the dependency tree for the sentence "*John found a solution to the problem*".



The links in the diagram represent dependency relationships. The direction of a link is from the head to the modifier in the relationship. Labels associated with the links represent types of dependency relations.

We define a collocation to be a dependency relationship that occurs more frequently than predicted by assuming the two words in the relationship are independent of each other. In [12], we described a method to create a collocation database by parsing a large corpus. Given a word *w*, the database can be used to retrieve all the dependency relationships involving *w* and the frequency counts of the dependency relationships. Table 1 shows excerpts of the entries in the collocation database for the words *duty* and *responsibility*. For example, in the corpus from which the collocation database is constructed, *fiduciary duty* occurs 319 times and *assume [the] responsibility* occurs 390 times.

The entry of a given word in the collocation database can be viewed as a feature vector for that word. Similarity between words can be computed using the feature vectors. Intuitively, the more features that are shared between two words, the higher the similarity between the two words. This intuition is captured by the Distributional Hypothesis [8].

[1] Available at www.cs.ualberta.ca/~lindek/demos.htm.

Table 2. The top 20 most similar words of *duty* and *eat* as given by (Lin, 1998b).

WORD	SIMILAR WORDS (WITH SIMILARITY SCORE)
DUTY	responsibility 0.182, obligation 0.138, job 0.127, post 0.121, function 0.121, task 0.119, role 0.116, assignment 0.114, requirement 0.109, tariff 0.109, mission 0.109, position 0.108, restriction 0.103, procedure 0.101, tax 0.101, salary 0.100, fee 0.099, training 0.097, commitment 0.096, penalty 0.095
EAT	cook 0.127, drink 0.108, consume 0.101, feed 0.094, taste 0.093, like 0.092, serve 0.089, bake 0.087, sleep 0.086, pick 0.085, fry 0.084, freeze 0.081, enjoy 0.079, smoke 0.078, harvest 0.076, love 0.076, chop 0.074, sprinkle 0.072, Toss 0.072, chew 0.072

Features of words are of varying degree of importance. For example, while almost any noun can be used as object of *include*, very few nouns can be modified by *fiduciary*. Two words sharing the feature *object-of-include* is less indicative of their similarity than if they shared the feature *modified-by-fiduciary*. The similarity measure proposed in [13] takes this into account by computing the mutual information between two words involved in a dependency relationship.

Using the collocation database, Lin [13] used an unsupervised method to construct a similarity matrix. Given a word *w*, the matrix returns a set of similar words of *w* along with their similarity to *w*. For example, the 20 most similar words of *duty* and *eat* are shown in Table 2.

4. Unsupervised Induction of Semantic Classes

The data in the collocation database can be viewed as a collection of feature vectors. Each unique word corresponds to a vector and each distinct dependency relationship that involves the word corresponds to a feature. For the newspaper corpus used in our experiments, we collected over 146,290 unique words and 20,173,092 features (1,639,996 unique). Although there are many clustering algorithms that take feature vectors as input, none seems to be able to handle the high dimensionality and the large number of elements.

4.1 CLIMAX

To deal with such a large set of data in a high-dimensional space, our approach is to break up the large set into many small (e.g. up to 20) subsets, which may overlap. We first use CLIMAX, a heuristic maximal-clique algorithm, to find clusters for each subset. We then apply the UNICON algorithm (Section 4.2) to merge and cluster the non-overlapping clusters returned by CLIMAX.

Figure 1 outlines the CLIMAX algorithm. In Step 1, a sequential greedy heuristic [3] is used to compute C_e since finding the maximum clique is an NP-complete problem [2]. Because we use a heuristic and, more importantly, since we limit the number elements to be clustered at any given point, the maximum-clique-based clustering algorithm can be executed efficiently. For example, in our experiments, the set E contained about 20,000 words. Step 1 took less than a minute to run on a Pentium III 700Mhz processor.

```
Input:   A similarity matrix $M$ for a list of elements $E$ and a
         number $n$.

Step 1:  Extract a set of cliques for each element. A set of
         elements is a clique if each element in the set is
         among the top $n$ most similar to each other.

         For each element $e \in E$
             $S_e \leftarrow e$ and top $n$ most similar elements to $e$
             $C_e \leftarrow$ collection of maximal cliques in $S_e$
         $C \leftarrow \bigcup_{e \in E} C_e$

Step 2:  Sort the collection of cliques in $C$ in descending
         order of clique size and average similarity among
         clique members.

Step 3:  Remove cliques in $C$ that have overlap with higher
         ranked cliques.

Output:  A list of clusters.
```

Figure 1.CLIMAX algorithm.

4.2 UNICON

The output of CLIMAX is a set of small clusters. Many of them are closely related. For example, two of the clusters returned by CLIMAX are:

```
(Nq34
  "Harvard University"            0.610996
  Harvard                         0.482834
  "Stanford University"           0.469302
  "University of Chicago"         0.454686
  "Columbia University"           0.44262
  "New York University"           0.436737
  "University of Michigan"        0.43055
  "Yale university"               0.416731
  MIT                             0.414907
  "University of Pennsylvania"    0.384016
  "Cornell University"            0.333958
)

(Nq184
  "University of Rochester"       0.525389
  "University of Miami"           0.466607
  "University of Colorado"        0.46347
  "Ohio State University"         0.430326
  "University of Florida"         0.398765
  "Harvard Medical School"        0.39485
  "University of North Carolina"  0.394256
  "University of Houston"         0.371618
)
```

Nq34 and *Nq184* are automatically generated names for the clusters. The number after each word in the clusters is the similarity between the word and the centroid of that cluster.

The UNICON algorithm computes the centroids of the clusters and merges clusters whose centroids are very similar. The sets of clusters to be merged are identified by applying the CLIMAX algorithm to the similarity matrix of the centroids.

Figure 2 outlines UNICON, our algorithm for unsupervised induction of concepts.

```
Input:   A collocation database $D$, a similarity matrix $M$ for
         a list of words $E$, and a number $n$.
Step 1:  $C \leftarrow$ CLIMAX($M$, $E$, $n$).
Step 2:  For each cluster $c \in C$, compute its centroid.
Step 3:  Compute the similarity matrix $M'$ between all
         centroids using (Lin, 1998b).
Step 4:  $S \leftarrow$ CLIMAX($M'$, $C$, $n$), where $S$ is a collection of
         subsets of $C$. Each subset is a cluster of clusters.
Step 5:  For each element in $S$, remove the clusters from $C$
         that belong to the element and create a new cluster
         in $C$ that is the union of these clusters.
Step 6:  Go to Step 2 unless $S$ is empty.
Step 7:  Compute the centroids of all the clusters in $C$ and
         add them as pseudo-words in $D$.
Step 8:  Find all the similar words, above a certain threshold,
         to the centroids using (Lin, 1998b) and add them to
         the corresponding clusters and store their similarity
         to the centroid.
Step 9:  For each cluster $c$ in $C$, remove each word from $c$
         that has a similarity
         For each word $w$ in $C$
         Find all clusters $w$ belongs to and the similarities to
             the clusters.
         Remove $w$ from clusters where its similarity is lower
             than 90% of the highest similarity among all
             clusters.
Output:  $C$, a list of clusters.
```

Figure 2. UNICON algorithm.

In Step 2, we compute the centroid of each cluster. A centroid consists of a set of features obtained by averaging the features of the words in the cluster. The averages are weighed by the frequency of the words so that the centroid can be treated as a pseudo-word. We can then filter out those features whose mutual information is below a fixed threshold (0.5 was used in our experiments).

Sometimes, a very frequent word within a cluster may hijack the centroid (i.e. there is hardly any influence on the centroid from all other words). Consider the following cluster in the output of CLIMAX:

```
{degree, master's degree, doctorate,
bachelor's degree, law degree, Ph.D., MBA,
M.B.A.}
```

The word *degree* is much more frequent than the other words. Consequently, the centroid for this cluster represents a mixture of all different meanings of *degree*. After removing the word *degree*, the regenerated centroid represents academic degrees and has a higher total similarity to the members in the cluster.

In Step 2, we recompute the centroid after all possible removals of members of a cluster. If the total similarity between the centroid and the members of the cluster decreases, the removed word is reinserted in the cluster. Then, in Step 3, the similarity matrix, *M*', is obtained by creating a new collocation database consisting of the centroids (pseudo-words). This matrix is used in Step 4 to determine which clusters should be merged.

After adding the centroids to the original collocation database in Step 7, we can use the features of the centroids to classify other words. The purpose of Step 8 is to expand the coverage of the clusters. For example, in one of our experiments, at the end of Step 6, the union of all clusters contains 5946 unique words. After Step 8, the number reaches 89,226. Many words added in Step 8 do not belong to a given cluster but nonetheless have a similarity to its centroid higher than a fixed threshold. For example, the word *pizza* is added to the cluster

```
{gin, bourbon, whiskey, vodka, rum, brandy,
scotch}
```

because it shares the following features with the cluster:

```
producer-of, market-for, object-of-buy,
object-of-enjoy, object-of-like, object-of-
market, object-of-order, object-of-sell,
object-of-serve, …
```

4.3 Discussion

4.3.1 Thresholding

One problem that plagued previous word similarity methods was that one always had to rely on an arbitrarily chosen threshold to determine similarity boundaries. In our approach, the responsibility of the threshold is reduced. If a very good cluster for a word *w* is found, it is used to remove membership of *w* from other clusters to which it is less similar. For example, *pizza* has the highest similarity to the cluster:

```
{sandwich, hamburger, hot dog}
```

Consequently, *pizza* is removed from the *hard liquor* cluster seen in the previous section.

4.3.2 Classifying unknown words

To classify a previously unknown word into a cluster, we simply have to compare the feature vector for this word with the cluster centroids. For example, the centroid for the cluster of news agencies,

```
{adn, Tanjug, PAP, CTK, Xinhua, MTI, Prensa
Latina, IRNA, Islamic Republic News Agency,
Xinhua News Agency, Tehran Radio, Notimex,
Yonhap, Press Association, Kyodo, Interfax,
Excelsior},
```

contains features such as

```
correspondent-for, object-of-monitor, object-
of-quote, subject-of-report, modified-by-
official, modifies-news-service, modifies-
commentary, modifies-news-agency, modified-
by-national, modified-by-state-run, …
```

When other words have many of these features, they can be classified as a news agency.

4.3.3 Handling low-frequency words

Since the centroid of a cluster is computed by averaging the frequency counts of features of the members, the more important features for the cluster tend to have higher frequency counts and mutual information. This makes it easier to deal with words that have a very small number of features. Consider the word *Venpres*[2]. It occurred in our corpus four times and has the following features:

```
subject-of-say, modified-by-news-agency,
modified-by-official, modified-by-Venezuelan,
modified-by-the
```

The top ten similar words of *Venpres* obtained by [13] are:

```
State Statistical Institute
State Statistics Bureau
Ministry for Economic Affairs
DPA
National Institute for Statistics
National Steel Manufacturers Assoc.
Bangladesh Sangbad Sangstha
Statistics Department
Central Bureau for Statistics
Orbe
```

Only two of the similar words, *DPA* and *Orbe*, are news agencies. The other words have high similarity to *Venpres* because they have the features s*ubject-of-say* and *modified-by-the*. When comparing the similarity between *Venpres* and all centroids, the most similar cluster is the *news agency* cluster shown in the previous subsection.

5. Experimental Results

In this section, we describe our test data and present an evaluation of our system.

5.1 Test Data and Experimental Set-up

We used two corpora for evaluating UNICON: *NEWS* consisting of a 1GB general newspaper corpus and *MEDLINE* consisting of a 54-million word corpus of Medline abstracts. We used Minipar[3], a descendent of Principar [14], to parse both corpora at a speed of about 500 words per second on a PIII-750 with 500MB memory. Table 3 shows the running-time of the algorithm (not including the construction time for the collocation database and word similarity matrix) and some results from our UNICON algorithm.

5.2 Evaluation of Centroids

We evaluate the capacity of the UNICON algorithm to classify words into existing clusters and the quality of the cluster centroids. We sorted the words in the *NEWS* and *MEDLINE* corpora that occurred a minimum of 100 times and selected every 150th (for *NEWS*) and every 75th (for *MEDLINE*) word. For each test word, we computed the similarity between it and all the clusters. Up to two of the most similar clusters were extracted. Each cluster is represented by up to 5 words in the cluster. We then gave the output to human judges for evaluation. Below is a sample of the test data given to judges.

```
Word:    bay
Cluster: {River lake creek ocean stream}

Word:    angel
Cluster: {Giants A's dodger warrior brave}
```

Table 3. Description of our experiments with the *NEWS* and *MEDLINE* corpus.

	NEWS	*MED*
Running-Time	4 hours	6.5 hours
Iterations of Steps 2 – 6 in Figure 3	6	5
Number of words after Step 6 in Figure 3	5927	7375
Number of words in the output clusters	89114	221655
Total number of clusters	1003	1094

Two judges inspected the *NEWS* corpus. For the *MEDLINE* corpus, two medical doctors performed the evaluation as a team. Each evaluator assigned a score between 1 and 5. Below, we describe each classification and provide an example.

1. The cluster is non-sensical and no determination for the test word may be made.
```
Word:    unevenness
Cluster: {Certfs, running time, cooling-off}
```
2. The test word does not fit well in the cluster.
```
Word:    Rose
Cluster: {Price, sale, growth, profit, rate}
```
3. Undecided
```
Word:    spray
Cluster: {oil, crude oil, gas gasoline,
         natural gas}
```
4. The test word fits with the general sense of the cluster.
```
Word:    inheritance
Cluster: {reimbursement, refund,
         compensation}
```
5. The test word fits perfectly with the cluster.
```
Word:    Seattle
Cluster: {St. Louis, Kansas City, Cleveland,
         Cincinnati, Pittsburgh}
```

The judges also had another category for unknown examples. Table 4 illustrates the result of the experiment. The average difference is the average absolute difference between each classification from the two human judges.

In our experiments, 913 concepts were induced by UNICON. Table 5 shows three of them. The first column contains the pseudo-words representing the names of the concepts and the third column shows the members of each concept. The members are listed in order of their similarity to the centroid of the clusters. For the sake of space, we omit the similarity values.

6. Conclusion and Future Work

We presented an unsupervised algorithm, UNICON, for inducing a set of concepts. Our system addresses some of the limitations of previous distributional approaches. The main contributions of our algorithm are:

- we output a set of concepts instead of a similarity matrix;
- we are able to deal with a large number of elements in a high-dimension space;
- we are able to classify words with few features; and

[2] Venpres is a Venezuelan news agency.

[3] Available at www.cs.ualberta.ca/~lindek/minipar.htm.

Table 4. Evaluation of cluster centroids. Two judges evaluated the *NEWS* corpus while a team of two medical doctors evaluated the *MEDLINE* corpus.

	NEWS		*MEDLINE*
	JUD. 1	*JUD. 2*	
Classification 1	3.2%	3.9%	6.8%
Classification 2	12.9%	11.0%	13.6%
Classification 3	0. 64%	3.9%	14.4%
Classification 4	16.8%	19.4%	18.2%
Classification 5	65.8%	58.1%	17.4%
Unknown Examples	0.64%	3.9%	29.5%
Average Score	4.30	4.21	3.37
Total Examples	**155**		**132**
Average Score	**4.26**		**3.37**
Average Difference	**0.34**		**N/A**

- we are able to classify previously unknown words into existing clusters.

We plan to use the automatically induced concepts to automatically generate verb usage templates. For example, for the word *express*, we may want to generate a template such as:

```
Nq23 expressed Nq45 about Nq4 in Nq198
```

where *Nq23* is a cluster of persons, *Nq45* is a cluster of feelings, *Nq4* is a cluster of events, and *Nq198* is a cluster of media.

UNICON does not generate a hierarchy among its output concepts. However, manual inspection of the output does show that there are many interesting semantic relationships among the clusters. For example, in the output of the *NEWS* corpus, there are clusters of general person names but also clusters of movie stars, U.S. senators and representatives, baseball players, names of well-known criminals, U.S. Justices, and government officials. Also, there are clusters of general company names, but we find clusters of high-tech companies, automakers, U.S. banks and international banks. It would be very interesting to automatically discover such relationships between these concepts.

7. ACKNOWLEDGMENTS

The authors wish to thank the reviewers for their helpful comments. This research was partly supported by Natural Sciences and Engineering Research Council of Canada grant OGP121338 and scholarship PGSB207797.

Table 5. Three concepts discovered by UNICON.

CONCEPT	MEMBERS
Nq178	Toyota, Honda, Volkswagen, Mazda, Oldsmobile, BMW, Audi, Mercedes-Benz, Cadillac, Volvo, Subaru, Chevrolet, Mercedes, Buick, Porsche, Nissan, VW, Mitsubishi, Renault, Hyundai, Isuzu, Jaguar, Suzuki, Dodge, Rolls-Royce, Pontiac, Fiat, Chevy, Saturn, Yugo, Ferrari, "Mercedes Benz", Plymouth, mustang, Beretta, Panasonic, Corvette, Nintendo, Camaro
Nq352	heroin, cocaine, marijuana, narcotic, alcohol, steroid, crack, opium
Nq356	Saskatchewan, Alberta, Manitoba, "British Columbia", Ontario, "New Brunswick", Newfoundland, Quebec, Guangdong, "Prince Edward Island", "Nova Scotia", "Papua New Guinea", "Northwest Territories", Luzon

8. REFERENCES

[1] Agrawal, R., Gehrke, J., Gunopulos, D., and Raghavan, P. 1998. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. In *Proceedings of ACM SIGMOD Conference on Management of Data*. pp. 94-105, Seattle, WA.

[2] Arora, S. and Sagra, S. 1992. Approximating Clique is NP-Complete. In *Proceedings of IEEE Symposium on Foundations of Computer Science*. pp. 2-13.

[3] Bomze, I. M., Budinich, M., Pardalos, P. M., and Pelillo, M. 1999. The maximum clique problem. *Handbook of Combinatorial Optimization (Supplement Volume A)*. D.-Z. Du and P. M. Pardalos (Eds.). Kluwer Academic Publishers. Boston, MA. pp. 1-74

[4] Chen, H., Schatz, B. R., Yim, T., and Fye, D. 1995. Automatic Thesaurus Generation for an Electronic Community System. *Journal of the American Society for Information Science*, 46(3):175-193.

[5] Crouch, C. J. and Yang, Bokyung. 1992. Experiments in Automatic Statistical Thesaurus Construction. In *Proceedings of SIGIR-92*. pp. 77-88.

[6] Grefenstette, G. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers.

[7] Harabagiu, S., Pasca, M. A., and Maiorano, S. J. 2000. Experiments with Open-Domain Textual Question Answering. In *Proceedings of COLING-2000*. pp. 292-298. Saarbrucken, Germany.

[8] Harris, Z. 1985. Distributional Structure. In: Katz, J. J. (ed.) *The Philosophy of Linguistics*. New York: Oxford University Press. pp. 26-47.

[9] Hays, D. 1964. Dependency Theory: a Formalism and Some Observations. *Language*, 40:511-525.

[10] Hudson, R. 1984. *Word Grammar*. Basil Blackwell Publishers Limited. Oxford, England.

[11] Jing, Y. and Croft, W. B. 1994. An Association Thesaurus for Information Retrieval. In *Proceedings of RIAO-94*. pp. 146-160. New York.

[12] Lin, D. 1998a. Extracting Collocations from Text Corpora. *Workshop on Computational Terminology*. pp. 57-63. Montreal, Canada.

[13] Lin, D. 1998b. Automatic Retrieval and Clustering of Similar Words. In *Proceedings of COLING-ACL98*. pp. 768-774. Montreal, Canada.

[14] Lin, D. 1994. Principar - an Efficient, Broad-Coverage, Principle-Based Parser. In *Proceedings of COLING-94*. pp. 42-48. Kyoto, Japan.

[15] McCallum, A., Nigam, K., and Ungar, L. H. 2000. Efficient Clustering of High-Dimensional Data Sets with Application to Reference Matching. In *Proceedings of KDD-2000*. Boston, MA.

[16] Mel'čuk, I. A. 1987. *Dependency Syntax: theory and practice*. State University of New York Press. Albany, NY.

[17] Park, Y. C. and Choi, K-S. 1996. Automatic Thesaurus Construction using Bayesian Networks. *Information Processing and Management*,32(5):543-553.

[18] Resnik, P. 1998. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of AI Research*, 11:95-130.

[19] Salton, G., Buckley, C., and Yu, C. T. 1983. An Evaluation of Term Dependence Models in Information Retrieval. *LNCS 146*. pp.151-173.

[20] Yangarber, R., Grishman, R., Tapanainen, P., and Huttunen, S. 2000. Automatic Acquisition of Domain Knowledge for Information Extraction. In *Proceedings of COLING-2000*. pp. 940-946. Saarbrucken, Germany.