Automatic Semantics Using Google

Rudi Cilibrasi*
CWI

Paul Vitanyi†
CWI, University of Amsterdam,
National ICT of Australia

Abstract

We have found a method to automatically extract the meaning of words and phrases from the world-wide-web using Google page counts. The approach is novel in its unrestricted problem domain, simplicity of implementation, and manifestly ontological underpinnings. The world-wide-web is the largest database on earth, and the latent semantic context information entered by millions of independent users averages out to provide automatic meaning of useful quality. We demonstrate positive correlations, evidencing an underlying semantic structure, in both numerical symbol notations and number-name words in a variety of natural languages and contexts. Next, we demonstrate the ability to distinguish between colors and numbers, and to distinguish between 17th century Dutch painters; the ability to understand electrical terms, religious terms, emergency incidents, and we conduct a massive experiment in understanding WordNet categories; the ability to do a simple automatic English-Spanish translation.

1 Introduction

Objects can be given literally, like the literal four-letter genome of a mouse, or the literal text of *War and Peace* by Tolstoy. For simplicity we take it that all meaning of the object is represented by the literal object itself. Objects can also be given by name, like "the four-letter genome of a mouse," or "the text of *War and Peace* by Tolstoy." There are also objects that cannot be given literally, but only by name and acquire their meaning from their contexts in background common knowledge in humankind, like "home" or "red." To make computers more intelligent one would like to represent meaning in computer-digestable form. Long-term and labor-intensive efforts like the *Cyc* project [12] and the *WordNet* project [21] try to establish semantic relations between common objects, or, more precisely, *names* for those objects. The idea is to create a semantic web of such vast proportions that rudimentary intelligence and knowledge about the real world spontaneously emerges. This comes at the great cost of designing structures capable of manipulating knowledge, and entering high quality contents in these structures by knowledgeable human experts. While the efforts are long-running and large scale, the overall information entered is minute compared to what is available on the world-wide-web.

The rise of the world-wide-web has enticed millions of users to type in trillions of characters to create billions of web pages of on average low quality contents. The sheer mass of the information available about almost every conceivable topic makes it likely that extremes will cancel and the majority or average is meaningful in a low-quality approximate sense. We devise a general method to tap the amorphous low-grade knowledge available for free on the world-wide-web, typed in by local users aiming at personal gratification of diverse objectives, and yet globally achieving what is effectively the largest semantic electronic database in the world. Moreover, this database is available for all by using search engines like Google.

Previously, we and others developed a compression-based method to establish a universal similarity metric among objects given literally as finite binary strings [15, 1, 16, 5]. Such objects can be genomes, music pieces in MIDI

*Supported in part by the Netherlands ICES-KIS-3 program BRICKS, and by NWO project 612.55.002. Address: CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands. Email: Rudi.Cilibrasi@cwi.nl.

†Part of this work was done while the author was on sabbatical leave at National ICT of Australia, Sydney Laboratory at UNSW. Supported in part by the EU EU Project RESQ IST-2001-37559, the ESF QiT Programmme, and the EU NoE PASCAL. Address: CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands. Email: Paul.Vitanyi@cwi.nl.

format, computer programs in Ruby or C, pictures in simple bitmap formats, or time sequences such as heart rhythm data. This precludes comparison of abstract notions or other objects that don't lend themselves to direct analysis, like emotions, colors, Socrates, Plato, Mike Bonanno and Albert Einstein. Here we develop a method that uses only the name of an object and obtains knowledge about the similarity of objects by tapping and distilling the great mass of available information on the web.

Intuitively, the approach is as follows. The Google search engine indexes around ten billion pages on the web today. Each such page can be viewed as a set of index terms. A search for a particular index term, say "horse", returns a certain number of hits (web pages where this term occurred), say 46,700,000. The number of hits for the search term "rider" is, say, 12,200,000. It is also possible to search for the pages where both "horse" and "rider" occur. This gives, say, 2,630,000 hits. This can be easily put in the standard probabilistic framework. If w is a web page and x a search term, then we write $x \in w$ to mean that Google returns web page w when presented with search term x. An *event* is a set of web pages returned by Google after it has been presented by a search term. We can view the event as the collection of all contexts of the search term, background knowledge, as induced by the accessible web pages for the Google search engine. If the search term is x, then we denote the event by $\mathbf{x}$, and define $\mathbf{x} = \{w : x \in w\}$. The *probability* $p(x)$ of an event $\mathbf{x}$ is the number of web pages, the *frequency* $f(x)$, in the event $\mathbf{x}$, divided by the overall number M of web pages possibly returned by Google. Thus, $p(x) = f(x)/M$. At the time of writing, Google searches $M = 8,058,044,651$ web pages. Define the joint event $\mathbf{x} \bigcap \mathbf{y} = \{w : x,y \in w\}$ as the set of web pages returned by Google, containing both the search term x and the search term y. The joint probability $p(x,y) = |\{w : x,y \in w\}|/M$ is the number of web pages in the joint event divided by the overall number M of web pages possibly returned by Google. This notation also allows us to define the probability $p(x|y)$ of *conditional* events $\mathbf{x}|\mathbf{y} = (\mathbf{x} \bigcap \mathbf{y})/\mathbf{y}$ defined by $p(x|y) = p(x,y)/p(y)$.

In the above example we have therefore $p(horse) \approx 0.0058$, $p(rider) \approx 0.0015$, $p(horse, rider) \approx 0.0003$. We conclude that the probability $p(horse|rider)$ of "horse" accompanying "rider" is $\approx 1/5$ and the probability $p(rider|horse)$ of "rider" accompanying "horse" is $\approx 1/19$. The probabilities are asymmetric, and it is the least probability that is the significant one. A very general search term like "the" occurs in virtually all (English language) web pages. Hence $p(the|rider) \approx 1$, and for almost all search terms x we have $p(the|x) \approx 1$. But $p(rider|the) \ll 1$, say about equal to $p(rider)$, and gives the relevant information about the association of the two terms.

Based on theoretical analysis [6], available to the reviewers on the web (but further invisible), and our previous work referred to above, we propose the following *normalized Google distance*

$$\mathrm{NGD}(x,y) = \frac{\max\{\log 1/p(x|y), \log 1/p(y|x)\}}{\max\{\log 1/p(x), \log 1/p(y)\}}. \tag{1}$$

with default values for the undefined cases. Note that $p(x|y) = p(x,y)/p(x) = 0$ means that the search terms "x" and "y" never occur together.

With the Google hit numbers above, we can now compute

$$\mathrm{NGD}(horse, rider) \approx 0.453.$$

We did the same calculation when Google indexed only one-half of the current number of pages: 4,285,199,774. It is instructive that the probabilities of the used search terms didn't change significantly over this doubling of pages, with number of hits for "horse" equal 23,700,000, for "rider" equal 6,270,000, and for "horse, rider" equal to 1,180,000. The $\mathrm{NGD}(horse, rider)$ we computed in that situation was 0.4445. This is in line with our contention that the relative frequencies of web pages containing search terms gives objective information about the semantic relations between the search terms. If this is the case, then with the vastness of the information accessed by Google the Google probabilities of search terms and the computed NGD 's should stabilize (be scale invariant) with a growing Google database.

2 Experiments

We verify that Google page counts capture something more than meaningless noise. For now we look at just the Google probabilities of small integers in several formats. The first format is the standard numeric representation using digits, for example "43". The next format is the number spelled out in English, as in "forty three". Then we use the

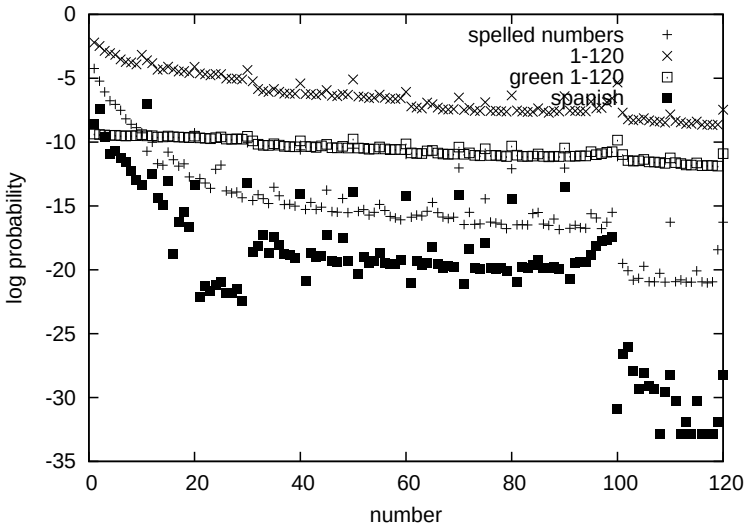

Figure 1: Numbers versus log probability (pagecount / M) in a variety of languages and formats.

number spelled in Spanish, as in “cuarenta y tres”. Finally, we use the number as digits again, but now paired with the fixed and arbitrary search term *green* . In each of these examples, we compute the probability of search term x as $f(x)/M$. We plotted $\log(f(x)/M)$ against x in Figure 1 for x runs from 1 to 120. Notice that numbers such as even multiples of ten and five stand out in every representation in the sense that they have much higher frequency of occurrence. We can treat only low integers this way, because there is not enough web pages to give significant hit counts to high integers.

Visual inspection of the plot gives clear evidence that there is a positive correlation between every pair of formats. We can therefore assume that that there is some underlying structure that is independent of the language chosen, and indeed the same structure appears even in the restricted case of just those webpages that contain the search term *green* .

3 Semantic Relations

We use the NGD where the objects to be related are search terms consisting of the names of colors, numbers, and some tricky words. Our program, arranging the objects in a tree visualizing the pairwise NGD’s, automatically organized the colors towards one side of the tree and the numbers towards the other, Figure 2. It arranges the terms which have as only meaning a color or a number, and nothing else, on the farthest reach of the color side and the number side, respectively. It puts the more general terms black and white, and zero, one, and two, towards the center, thus indicating their more ambiguous interpretation. Also, things which were not exactly colors or numbers are also put towards the center, like the word “small”. We may consider this an example of automatic ontology creation.

In the example of Figure 3, the names of fifteen paintings by Steen, Rembrandt, and Bol were entered. The names of the associated painters were not included in the input, however they were added to the tree display afterward to demonstrate the separation according to painters. The paintings used are: **Rembrandt van Rijn** : *Hendrickje slapend; Portrait of Maria Trip; Portrait of Johannes Wtenbogaert ; The Stone Bridge ; The Prophetess Anna* ; **Jan Steen** : *Leiden Baker Arend Oostwaert ; Keyzerswaert ; Two Men Playing Backgammon ; Woman at her Toilet ; Prince’s Day ; The Merry Family* ; **Ferdinand Bol** : *Maria Rey ; Consul Titus Manlius Torquatus ; Swartenhont ; Venus and Adonis* This type of problem has attracted a great deal of attention [10]. A more classical solution is offered in[11], where a domain-specific database is used for similar ends. The present automatic oblivious method obtains results that compare favorably with the latter feature-driven method.

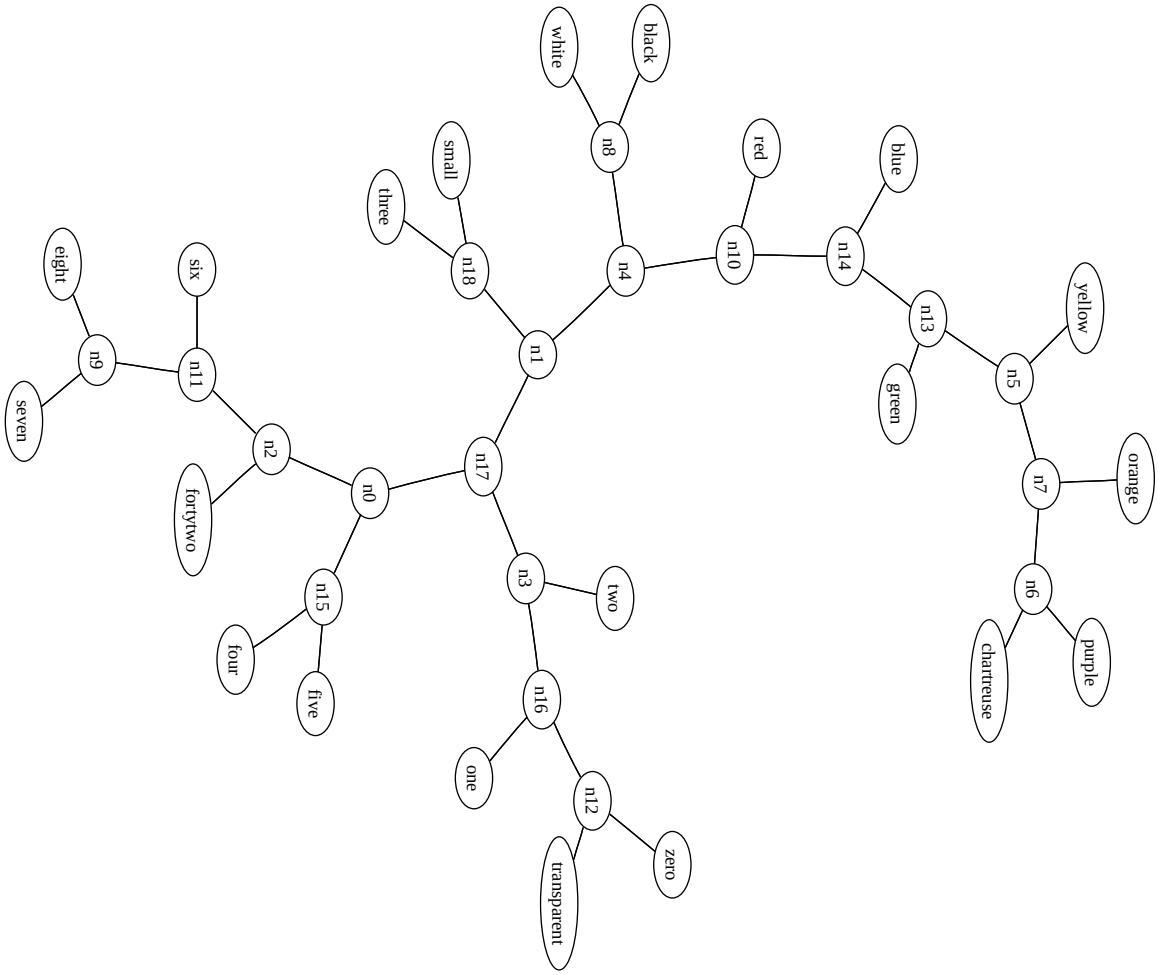

Figure 2: Colors and numbers arranged into a tree using NGD .

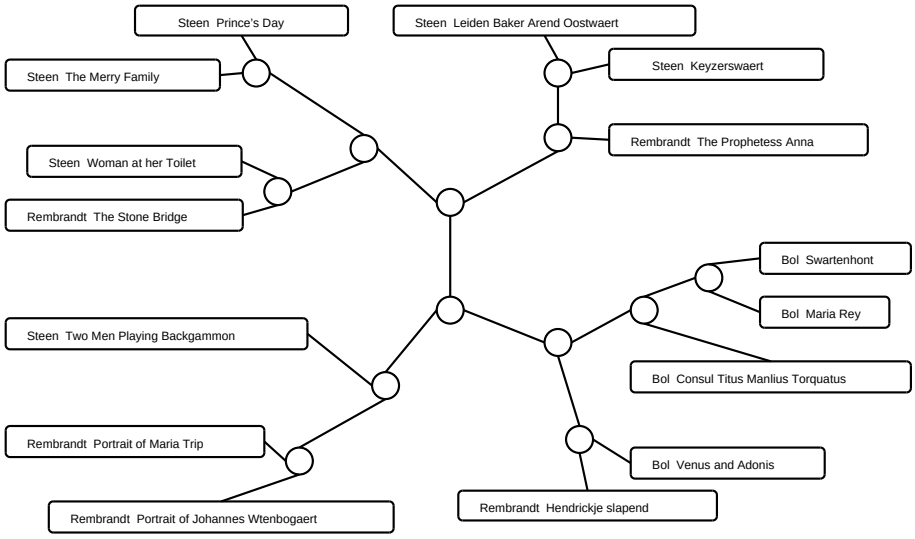

Figure 3: Fifteen paintings tree by three different painters

Training Data

Positive Training	(22 cases)			
avalanche	bomb threat	broken leg	burglary	car collision
death threat	fire	flood	gas leak	heart attack
hurricane	landslide	murder	overdose	pneumonia
rape	roof collapse	sinking ship	stroke	tornado
train wreck	trapped miners			
Negative Training	(25 cases)			
arthritis	broken dishwasher	broken toe	cat in tree	contempt of court
dandruff	delayed train	dizziness	drunkenness	enumeration
flat tire	frog	headache	leaky faucet	littering
missing dog	paper cut	practical joke	rain	roof leak
sore throat	sunset	truancy	vagrancy	vulgarity
Anchors	(6 dimensions)			
crime	happy	help	safe	urgent
wash				

Testing Results

	Positive tests	Negative tests
Positive Predictions	assault, coma, electrocution, heat stroke, homicide, looting, meningitis, robbery, suicide	menopause, prank call, pregnancy, traffic jam
Negative Predictions	sprained ankle	acne, annoying sister, campfire, desk, mayday, meal
Accuracy	15/20 = 75.00%	

Figure 4: Google- SVM learning of "emergencies."

4 Learning

Next, we augment the Google method by adding a trainable learning system. Here we use the Support Vector Machine (SVM) as a trainable component (we could also have used neural networks, but the SVM 's are simpler). The setting is a binary classification problem on examples represented by search terms. We require a human expert to provide a list of at least 40 *training words*, consisting of at least 20 positive examples and 20 negative examples, to illustrate the contemplated concept class. The expert also provides, say, six *anchor words* $a_1,\ldots,a_6$, of which half are in some way related to the concept under consideration. Then, we use the anchor words to convert each of the 40 training words $w_1,\ldots,w_{40}$ to 6-dimensional *training vectors* $\bar{v}_1,\ldots,\bar{v}_{40}$. The entry $v_{j,i}$ of $\bar{v}_j = (v_{j,1},\ldots,v_{j,6})$ is defined as $v_{j,i} = \mathrm{NGD}(w_i,a_j)$ ($1 \le i \le 40$, $1 \le j \le 6$). The training vectors are then used to train an SVM to learn the concept, and then test words may be classified using the same anchors and trained SVM model.

In Figure 4, we trained using a list of emergencies as positive examples, and a list of "almost emergencies" as negative examples. The figure is self-explanatory. The accuracy on the test set is 75%.

In Figure 5 the method learns to distinguish prime numbers from non-prime numbers by example:

The prime numbers example illustrates several common features of our method that distinguish it from the strictly deductive techniques. It is common for our classifications to be good but imperfect, and this is due to the unpredictability and uncontrolled nature of the Google distribution.

To create the next example, we used WordNet. WordNet is a semantic concordance of English, [21]. It also attempts to focus on the meaning of words instead of the word itself. The category we want to learn, the concept,

Training Data

Positive Training	(21 cases)			
11	13	17	19	2
23	29	3	31	37
41	43	47	5	53
59	61	67	7	71
73				
Negative Training	(22 cases)			
10	12	14	15	16
18	20	21	22	24
25	26	27	28	30
32	33	34	4	6
8	9			
Anchors	(5 dimensions)			
composite	number	orange	prime	record

Testing Results

	Positive tests	Negative tests
Positive Predictions	101, 103, 107, 109, 79, 83, 89, 91, 97	110
Negative Predictions		36, 38, 40, 42, 44, 45, 46, 48, 49
Accuracy	18/19 = 94.74%	

Figure 5: Google- SVM learning of primes.

is termed “electrical”, and represents anything that may pertain to electronics, Figure 6. The negative examples are constituted by simply everything else. The accuracy on the test set is 100%: It turns out that “electrical terms” are unambiguous and easy to learn and classify by our method.

In the next example, Figure 7, the concept to be learned is “religious”. Here the positive examples are terms that are commonly considered as pertaining to religious items or notions, the negative examples are everything else. The accuracy on the test set is 88.89%. Religion turns out to be less unequivocal and unambiguous than “electricity” for our method.

Notice that what we may consider to be errors, can be explained, or point at, a secondary meaning or intention of these words. For instance, some may consider the word “shepherd” to be full of religious connotation. Next, we estimated how well the Google method agrees with WordNet in a large number of automatically selected semantic categories. We ran 100 experiments. The actual data are available at [3]. A histogram of agreement accuracies is shown in Figure 8. On average, our method turns out to agree well with the WordNet semantic concordance made by human experts. The mean of the accuracies of agreements is 0.8725. The variance is ≈ 0.01367, which gives a standard deviation of ≈ 0.1169. Thus, it is rare to find agreement less than 75%.

5 Translation of Natural Languages

Another potential application of the NGD method is in natural language translation.

Assume that the system has already determined five English and five Spanish words that match, but the permutation associating the English and Spanish words is, as yet, undetermined. Using the NGD relations with a set of English and Spanish words of which the correct matching is known, the computer inferred the correct permutation for the testing words, see Figure 10.

6 Conclusion

A comparison can be made with the *Cyc* project [12]. Cyc, a project of the commercial venture Cycorp, tries to create artificial common sense. Cyc’s knowledge base consists of hundreds of microtheories and hundreds of thousands of terms, as well as over a million hand-crafted assertions written in a formal language called CycL [18]. CycL is an enhanced variety of first-order predicate logic. This knowledge base was created over the course of decades by paid human experts. It is therefore of extremely high quality. Google, on the other hand, is almost completely unstructured, and offers only a primitive query capability that is not nearly flexible enough to represent formal deduction. But what it lacks in expressiveness Google makes up for in size; Google has already indexed more than eight billion pages and shows no signs of slowing down. We have demonstrated that NGD can be used to extract meaning in a variety of ways from the statistics inherent to the Google database. So far, all of our techniques look only at the page count portion of the Google result sets and achieve surprising results. How much more amazing might it be when the actual contents of search results are analyzed? Consider the possibility of using WordNet familiarity counts to filter returned search results to select only the least familiar words, and then using these in turn as further inputs to NGD to create automatic discourse or concept diagrams with arbitrary extension. Or perhaps this combination can be used to expand existing ontologies that are only seeded by humans.

Acknowledgments

We thank Teemu Roos, Hannes Wettig, Petri Mylliymaki, and Henry Tirri at COSCO and The Helsinki Institute for Information Technology for interesting discussions. We also thank Chih-Jen Lin for providing, free of charge to all, the very easy to use LibSVM package. We thank the Cognitive Science Laboratory at Princeton University for providing the excellent and free WordNet database. And we wish to thank the staff of Google, Inc. for their support of this research by providing an API as well as generous access to their websearch system.

Training Data

Positive Training	(58 cases)			
Cottrell precipitator	Van de Graaff generator	Wimshurst machine	aerial	antenna
attenuator	ballast	battery	bimetallic strip	board
brush	capacitance	capacitor	circuit	condenser
control board	control panel	distributer	electric battery	electric cell
electric circuit	electrical circuit	electrical condenser	electrical device	electrical distributor
electrical fuse	electrical relay	electrograph	electrostatic generator	electrostatic machine
filter	flasher	fuse	inductance	inductor
instrument panel	jack	light ballast	load	plug
precipitator	reactor	rectifier	relay	resistance
security	security measures	security system	solar array	solar battery
solar panel	spark arrester	spark plug	sparking plug	suppresser
transmitting aerial	transponder	zapper		
Negative Training	(55 cases)			
Andes	Burnett	Diana	DuPonts	Friesland
Gibbs	Hickman	Icarus	Lorraine	Madeira
Quakeress	Southernwood	Waltham	Washington	adventures
affecting	aggrieving	attractiveness	bearer	boll
capitals	concluding	constantly	conviction	damming
deeper	definitions	dimension	discounting	distinctness
exclamation	faking	helplessness	humidly	hurling
introduces	kappa	maims	marine	moderately
monster	parenthesis	pinches	predication	prospect
repudiate	retry	royalty	shopkeepers	soap
sob	swifter	teared	thrashes	tuples
Anchors	(6 dimensions)			
bumbled swimmers	distributor	premeditation	resistor	suppressor

Testing Results

	Positive tests	Negative tests
Positive Predictions	cell, male plug, panel, transducer, transformer	
Negative Predictions		Boswellizes, appointer, enforceable, greatness, planet
Accuracy	10/10 = 100.00%	

Figure 6: Google- SVM learning of “electrical” terms.

Training Data

Positive Training	(22 cases)			
Allah	Catholic	Christian	Dalai Lama	God
Jerry Falwell	Jesus	John the Baptist	Mother Theresa	Muhammad
Saint Jude	The Pope	Zeus	bible	church
crucifix	devout	holy	prayer	rabbi
religion	sacred			
Negative Training	(23 cases)			
Abraham Lincoln	Ben Franklin	Bill Clinton	Einstein	George Washington
Jimmy Carter	John Kennedy	Michael Moore	atheist	dictionary
encyclopedia	evolution	helmet	internet	materialistic
minus	money	mouse	science	secular
seven	telephone	walking		
Anchors	(6 dimensions)			
evil	follower	history	rational	scripture
spirit				

Testing Results

	Positive tests	Negative tests
Positive Predictions	altar, blessing, communion, heaven, sacrament, testament, vatican	earth, shepherd
Negative Predictions	angel	Aristotle, Bertrand Russell, Greenspan, John, Newton, Nietzsche, Plato, Socrates, air, bicycle, car, fire, five, man, monitor, water, whistle
Accuracy	24/27 = 88.89%	

Figure 7: Google- SVM learning of “religious” terms.

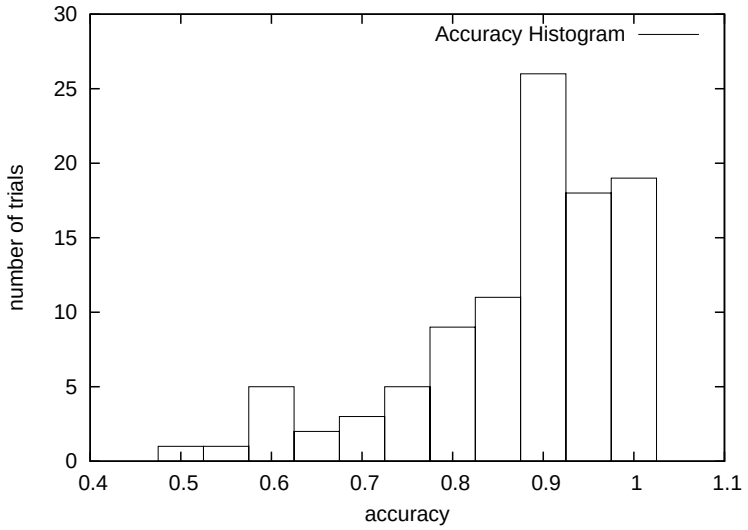

Figure 8: Histogram of accuracies over 100 trials of WordNet experiment.

Given starting vocabulary	
English	**Spanish**
tooth	diente
joy	alegria
tree	arbol
electricity	electricidad
table	tabla
money	dinero
sound	sonido
music	musica
Unknown-permutation vocabulary	
plant	bailar
car	hablar
dance	amigo
speak	coche
friend	planta

Figure 9: English-Spanish Translation Problem

	English	**Spanish**
Predicted (optimal) permutation	plant	planta
	car	coche
	dance	bailar
	speak	hablar
	friend	amigo

Figure 10: Translation Using NGD

References

[1] C.H. Bennett, P. Gács, M. Li, P.M.B. Vitányi, W. Zurek, Information Distance, *IEEE Trans. Information Theory*, 44:4(1998), 1407–1423.

[2] C.J.C. Burges. A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery*, 2:2(1998),121–167.

[3] Automatic Meaning Discovery Using Google: 100 Experiments in Learning WordNet Categories, 2004, `http://www.cwi.nl/~cilibrar/googlepaper/appendix.pdf`

[4] R. Cilibrasi, R. de Wolf, P. Vitanyi. Algorithmic clustering of music, *Computer Music Journal*, 2004.

[5] R. Cilibrasi, P. Vitanyi. Clustering by compression, Submitted to *IEEE Trans. Information Theory*. http://www.archiv.org/abs/cs.CV/0312044

[6] R. Cilibrasi, P. Vitanyi, Automatic meaning discovery using Google, full version, http://www.cwi.nl/~paulv/amdug.pdf

[7] The basics of Google search, http://www.google.com/help/basics.html.

[8] L.G. Kraft, A device for quantizing, grouping and coding amplitude modulated pulses. Master's thesis, Dept. of Electrical Engineering, M.I.T., Cambridge, Mass., 1949.

[9] L. Lakshmanan, F. Sadri. Xml interoperability, *Proc. Intn'l Workshop Web and Databases (WebDB)* San Diego, California, June 2003.

[10] L. Rutledge, M. Alberink, R. Brussee, S. Pokraev, W. van Dieten, M. Veenstra. Finding the Story — Broader Applicability of Semantics and Discourse for Hypermedia Generation, *Proc. 14th ACM Conf. Hypertext and Hypermedia* Nottingham, UK, pp. 67-76, August 23-27, 2003

[11] M.J. Alberink, L.W. Rutledge, M.J.A. Veenstra, Clustering semantics for hypermedia presentation, *CWI Tech Report*, INS-E0409, ISSN 1386-3681, 2004.

[12] Douglas B. Lenat. Cyc: A large-scale investment in knowledge infrastructure, *Comm. ACM*, 38:11(1995),33–38.

[13] A.N. Kolmogorov. Three approaches to the quantitative definition of information, *Problems Inform. Transmission*, 1:1(1965), 1–7.

[14] A.N. Kolmogorov. Combinatorial foundations of information theory and the calculus of probabilities, *Russian Math. Surveys*, 38:4(1983), 29–40.

[15] M. Li, J.H. Badger, X. Chen, S. Kwong, P. Kearney, and H. Zhang, An information-based sequence distance and its application to whole mitochondrial genome phylogeny, *Bioinformatics*, 17:2(2001), 149–154.

[16] M. Li, X. Chen, X. Li, B. Ma, P. Vitanyi. The similarity metric, *IEEE Trans. Information Theory*, 50:12(2004), 3250- 3264.

[17] M. Li, P. M. B. Vitanyi. *An Introduction to Kolmogorov Complexity and Its Applications*, 2nd Ed., Springer-Verlag, New York, 1997.

[18] S. L. Reed, D. B. Lenat. Mapping ontologies into cyc. *Proc. AAAI Conference 2002 Workshop on Ontologies for the Semantic Web*, Edmonton, Canada. http://citeseer.nj.nec.com/509238.html

[19] D.H. Rumsfeld, The digital revolution, originally published June 9, 2001, following a European trip. In: H. Seely, The Poetry of D.H. Rumsfeld, 2003, http://slate.msn.com/id/2081042/

[20] C. E. Shannon. A mathematical theory of communication. *Bell Systems Technical J.*, 27(1948), 379–423 and 623–656.

[21] G.A. Miller et.al, WordNet, A Lexical Database for the English Language, Cognitive Science Lab, Princeton University, http://www.cogsci.princeton.edu/ wn