Very Large Lexicons

Gregory Grefenstette

Clairvoyance Corp., Pittsburgh, USA

Abstract

Three independent phenomena have appeared over the last fifteen years that make it possible to think about linguistics in a different way. The first is the appearance of the World Wide Web; the second is the development of robust and rapid, though shallow, linguistic analyzers, and the third is the availability of cheap memory. In this chapter, we show how these three phenomena come together in the creation of a new linguistic resource: the Very Large Lexicon.

1 Introduction

Linguists are used to building models of language by hand. But the presence of a large amount of freely available text, coupled with robust parsers for treating it, gives us a new way of looking at language models. We can now consider things on a massive scale. Just as the four-color problem (Tymoczko 1990) was solved using a battery of computers, and as computer scientists are looking once again at brute force methods (Nievergelt, Gasser, Maeser and Wirth 1995), we can imagine treating linguistics problems in the same way. As a tool for such an approach, in this chapter, we sketch the idea of a Very Large Lexicon.

First we describe the existing natural language processing tools that allow a computer to work with text by abstracting away surface differences. Then we discuss how big the Web is and what language model we can extract from it. We examine whether we could store such a model. Finally we outline what would be included in a Very Large Lexicon and describe one possible application of it.

2 Phenomenon One – Natural Language Processing

The computer is a very simple tool. Essentially, it can only test for equality between two sequences of bits. (To be fair, it can also tell if one number is greater or less than another number. But this is really all a computer can do.) In order to exploit the power of the computer for the tasks we want it to perform, we must reduce things to some format that can be exactly compared. We, as humans, have innate abilities to see similarities: given two photos of the same tree from slightly different angles, we could still tell it was the same because we can abstract away differences. In the same way, with respect to text, we can see the three statements in Figure 1 as instances of some concept like ”steal painting” although the surface forms of the expressions are different.

In order to make make two different pieces of text look exactly alike to the computer, Computational Linguistics has created a number of models of language that map different strings of text or phonemes into identical forms.

...three paintings were stolen ...

... accused of stealing three paintings

... paintings they were about to steal...

Figure 1: We can abstract away the surface of details to see each of these text segments of variants of ”steal a painting”

There are many important applications of this action of abstracting away details. Classification systems (Yang and Pedersen 1997) ignore most of a document in order to reduce it to one of a number of predetermined classes (e.g., medical, financial, etc.). Information retrieval systems (Frakes and Baeza-Yates 1992) abstract away many aspects of a document by indexing words and phrases, reducing the document to a normalized list, so that the computer can match it to a query, reduced to a similar list. Information extraction systems (Robert Gaizauskas 1997) use models of events (such as company mergers) in which things irrelevant to the event are ignored and from which patterns corresponding to the event are recognized. Terminology recognition and control (Jacquemin 1999) uses a model of correct and incorrect terminology and the variations that they can undergo, so that technical text can be more precisely written. Text mining (Ghani, Jones, Mladenic, Nigam and Slattery n.d.) converts running text into abstracted lists and patterns in order to discover recurring patterns and combinations over a number of texts. Translating programs (Hutchins and Somers 1992) use models of grammar that map grammatical structures from one language to another, and models of the lexicon that use limited context to perform proper word choice in the target language.

In order to perform these abstractions, Natural Language Processing (NLP) uses a number of remapping models, either built by hand or derived from statistical analysis that allow the computer to see two different sequences as being the same. Examples of these models are:

- Speech language models (Placeway, Schwartz, Fung and Nguyen 1993): use models of real words in order to map phonetic strings into readings of words.

- Lexicons (Chanod and Tapanainen 1995): map different word forms to the same normalized form. For example *thinks, thinking* and *thought* are mapped to *think (verb)*. Stemming (Porter 1980) is sometimes used as a quick-and-dirty lexical normalization.

- Part-of-Speech taggers (Church 1988): use models of known sequences of grammatical categories (given by the lexicons) to choose between possible readings of sentences. Applying these models allows the computer to see abstract categories such as nouns, verbs, and adjectives in the place of real, different words.

- Grammars (Pereira and Warren 1980): use models of part-of-speech groupings that are mapped into higher-level structure, such noun phrases an verb

```
[ ]:LEFTCONTEXT
            token
                [ ]:MIDDLE
                        token
                           relation:[ ]
                                [ ]:RIGHTCONTEXT
```

Figure 2: Shallow parsers sometimes use regular expressions for modeling context and for transducing this recognized context to the empty string (written here as *[]*), while retaining in the output the tokens, and introducing in the output the recognized relation (written here as relation matching the empty string on input *relation:[]*).

phrases, that allow the computer to recognize two different sentences as, for example, being transitive uses of the verb *steal*.

2.1 Robust, Shallow Parsing

Shallow parsers (Grefenstette 1999) can extract a syntactic dependency between two words by modeling, for example using regular expressions (Karttunen, Chanod, Grefenstette and Schiller 1996), the structure of the syntactic context to the left of the first word (with, for example as seen in Figure 2, a regular expression `LEFTCONTEXT`), between the words (with a regular expression `MIDDLE`), and to the right of the last word (with a regular expression `RIGHTCONTEXT`). None of this context is output (it removed by mapping it to the empty string, `[ ]`) by the transducing filter as the context is recognized. The only items that are output by the filtering transducer are the tokens (the tagged words) that are found surrounded by the contexts, and a relation label inserted after the last token. Schematically this gives the structure in Figure 2.

These models, when applied to text, along with other models of lexical normalization and part of speech tagging, will produce abstracted versions of the syntactic relations that are found. Figure 3 shows how an input sentence is abstracted to a collection of dependency relations between the words in the sentence.

The last twenty years has seen the creation of a large number of shallow, robust parsers (Voutilainen, Heikkila and Anttila 1992, Hindle 1993, Debili 1982, Abney 1991, Ejerhed and Church 1983, Jensen, Heidorn and Richardson 1993, Ait-Mokhtar and Chanod 1997) able to extract syntactic relations quickly from large quantities of texts. Nuria Gala (Pavia 2000) is working on producing a shallow parser whose results can be assured to have a high precision.

3 Phenomenon Two – The World Wide Web

Natural Language Processing now provides us with tools (lexicons, part-of-speech taggers, grammars, parsers) from which we can derive an abstracted model of how language is used. The World Wide Web provides us with the text from which to

36 Helmantel paintings were stolen at this burglary.

[SC [NP 36 Helmantel paintings NP]/SUBJ :v were stolen SC] [PP at this burglary PP] .

ADJ(36,painting)

SUBJPASS(painting,steal)

NN(Helmantel,painting)

VMODOBJ(steal,at,burglary)

Figure 3: Shallow parsers sometimes produce a list of syntactic relations sing their models of these relations. A first pass on part-of-speech tagged text *(part-of-speech tags not shown here)* introduces more structure that is used by the syntactic relations models. Here a model of adjective modifiers produces ADJ, a syntactic relation between *36* and *painting*; Others models: a passive subject model (SUBJPASS), a noun modifier model (NN) and a verb modifier model (VMODOBJ) produce other abstracted versions of the sentence.

extract the models.

Everyone knows that the World Wide Web is huge. In order to give some idea of its size, Figure 4 gives the frequency of some random phrases on the Web and in the largest constructed corpus of English, the British National Corpus (see the URL info.ox.ac.uk/bnc/ for more information on this corpus). For example, we see that the ordinary phrase “deep breath” appears 374 times per 100 million words in the British National Corpus, but appeared in texts indexed by Altavista more than 79,000 times in June of 2001. These numbers show that the WWW is orders of magnitudes larger than this large corpus, and growing.

Lawrence and Giles estimated (Lawrence and Giles 1999) that the publicly indexable web contained about 800 million pages as of February 1999, or about 6 terabytes of text after removing the HTML. In June of 2001, Google’s homepage invited its users to search among its 1,346,966,000 web pages.

Using techniques described in (Grefenstette and Nioche 2000), we estimated that Altavista allows access to over 75 billion words of English in March, 2001, and many billions of words of other languages, The complete estimates are shown in Figure 5. Lawrence and Giles calculated that Altavista only indexes about 15% of the web, so the numbers given in Figure 5 must be considered as lower bounds on the amount of text available on the WWW.

The WWW is clearly big, but is it a good place to derive a language model from. It is not as clean as the corpora of newspaper texts that computational linguists are used to working with. Erros can arise from a number of sources: the people using the Internet may be writing their texts in a non-native language; they may be using incorrect speech; they will be making grammatical and spelling errors. How can we propose to derive a model of how language is used from the Web?

To allay these fears, we can make some anecdotal observations. Figures 6 and

	BNC	WWW	WWW
sample phrases	100 M Words	1999	2001
medical treatment	202	46064	342155
prostate cancer	28	40772	473210
deep breath	374	54550	79440
acrylic paint	20	7208	22288
perfect balance	28	9735	30077
electromagnetic radiation	24	17297	57421
powerful force	54	17391	32663
concrete pipe	8	3360	16737
upholstery fabric	5	3157	7008

Figure 4: Comparison of the frequency of some random English noun phrases in the British National Corpus and in Altavista in 1999 and in Altavista in 2001

7 show some common grammatical errors in Spanish and Dutch. The Spanish errors are called *dequeismos*, which means to place a spurious *de* between a verb and its following relative clause. The Dutch examples show improper choice of prepositions. Examples are easy to generate for English, also. For example, in June 2001, there were 1692 “I beleave”, 41617 “I beleive” but 3,800,810 “I believe.” In all these cases the number of correct forms is much greater than the number of erroneous forms. This seems to indicate that the WWW can be a source for modeling language if thresholds are applied. The Web is so big that the signal (correct forms and correct usage) is so strong and noise can be ignored.

4 Phenomenon Three – Disk Space

In the last two sections we have talked about two phenomema: Natural Language Processing and robust parsing, and the WWW and its size. Their conjunction offers promise that we can build a large model of how language is used, and build this for many written languages. But how large will this model be, and will we be able to store it?

First we can begin by deciding which words to model. With the NLP tools we have, we can generate all the surface forms for a language (up to a given character length). WWW browsers (e.g. Altavista, Alltheweb) allows us to rank these surface forms by frequency. So, we can consider the 200,000 most frequent surface forms for a language. Suppose that we want to build a very simple language model consisting of a two dimensional matrix of the relative collocation frequency of one word to another. Suppose that we store this frequency in 4 bytes. This means we need 200,000 by 200,000 by 4 bytes, or 160 gigabytes. (We would actually need much less, since the matrix would be very sparse, and could be stored using a more efficient representation.)

Though such a size for a model may still seem daunting, disk drives are getter bigger and cheaper. In June, 2001, disk space cost about $ 3 per gigabyte. By Moore’s Law (Schaller 1997) of computer power doubling every 18 months, and

	Word count estimate
Welsh	14,993,000
Albanian	10,332,000
Breton	12,705,000
Lithuanian	35,426,000
Latvian	39,679,000
Esperanto	57,154,000
Basque	55,340,000
Latin	55,943,000
Estonian	98,066,000
Irish	88,283,000
Icelandic	53,941,000
Roumanian	86,392,000
Croation	136,073,000
Slovenian	119,153,000
Turkish	187,356,000
Malay	157,241,000
Catalan	203,592,000
Slovakian	216,595,000
Finnish	326,379,000
Danish	346,945,000
Polish	322,283,000
Hungarian	457,522,000
Czech	520,181,000
Norwegian	609,934,000
Dutch	1,063,012,000
Swedish	1,003,075,000
Portuguese	1,333,664,000
Italian	1,845,026,000
Spanish	2,658,631,000
French	3,836,874,000
German	7,035,850,000
English	76,598,718,000

Figure 5: Estimates of number of words of text available for some languages on the WWW through Altavista in March, 2001.

	www.alltheweb.com (June 2001)
”pienso de que”	171 times
”pienso que”	83966 times
”piensas de que”	89 times
”piensas que”	11485 times
”piense de que”	9 times
”piense que”	12867 times
”pensar de que”	716 times
”pensar que”	188508 times

Figure 6: Frequency of pages containing ’dequeismos errors’ (placing a spurious ’de’ between the verb and the relative) on the Web. The correct cases appear two orders of magnitude more often.

	www.alltheweb.com (October 2000)
”hun hebben het”	10 times
”ze hebben het”	2459 times
”groter als”	1079 times
”groter dan”	20421 times
”betreffende hen”	12 times
”betreffende hun”	329 times
”behalve hen”	12 times
”behalve hun”	310 times

Figure 7: Frequency of some Dutch preposition choice errors on the Web: the erroneous cases appear much less often than the correct cases.

TREC cross-language query: Welke mogelijkheden zijn er voor hergebruik van afval?

afval	40494
mogelijkheden	198060
van	30169524
hergebruik	12397
welke	388139
er	2313010
zijn	5041618
voor	7958353

Welke mogelijkheden zijn er voor hergebruik van afval?

Figure 8: Word counts of Dutch words from the Web can be used to weight words in an informational retrieval system. In the example above, the original query about recycling garbage is represented with word size corresponding roughly to the weights derivable from inverse WWW frequency.

growing tenfold every five years, we can predict that a terabyte (1000 gigabytes), which cost $ 3000 in 2001, will cost $ 300 in 2006 and $ 30 in 2011. The whole storage market is being driven by the demand for cheap video and music. Storage of text models will always be very small compared to these media, and very soon we should be able to effectively store such large models cheaply.

5 The Very Large Lexicon

The convergence of these three phenomena (storage, NLP and the WWW) means that we can consider building a new linguistic resource. This resource, massive, but automatically derivable, is not what we are used to dealing with in NLP. What we propose is more than a simple lexicon containing all the forms of words, more than a list of idiomatic expressions, more than a large model of word pairs, more than a grammar, more than a dictionary made for humans. We propose storing an abstracted expression, derived from shallow parsing and other current NLP tools, of how lexical items are really used. In this section we will consider what this Very Large Lexicon should contain. As a minimum, we think it should store:

- relative frequencies of words
- co-occurrence patterns, and their frequencies
- dependency relations between words

5.1 Relative frequency of words

Many web portals give word and page counts for the queries users send. By generating queries consisting of all the word forms of a language, we can easily gather the word frequencies of the basic lexical items of a language. (There is one caveat: web portals do not distinguish between languages in these counts so that, for example, English and German ’die’ counts are conflated. There are ways to overcome these effects by referring to expected frequencies from smaller known-language corpora.)

Knowing the relative frequencies of words is useful for many NLP tasks: i.e. information retrieval system suppose that the word frequency is an indication of the importance of words, see Figure 8. These web frequencies can be used just as real corpus frequencies are used in closed-corpus information retrieval systems.

5.2 Collocation frequencies

If we know the frequencies of each word, and if we know the co-occurrence frequencies of the words, we can build into the Very Large Lexicons the list of the words with highest mutual information for each word. As an example, we took the words ’thief’ and ’piano’ and for each word we generated co-occurrence queries with all the other words in an English full-form lexicon using the NEAR operator of the Altavista advanced search option. Calculating mutual information (Church and Hanks 1990), we get the associated word lists shown in Figure 9. These groups

thief*: accused adventure alarm arch armor arrest assassination attack attempt beggar bicycle blessed break brothers burglary capture car catch chances character chase cheats clerical climbing come con conviction cook cop crack crime criminal cry cryptographic dagger damn dangerous dark destroy detection devil disappeared discovery doctors dragon dream druid dwarf elf enemy escape evil excite faith fight fled fool gentleman grab guard guild guilty guns guy hack hang happy hash healing heaven hero hidden honest honors horse hunting intrusion jewelry kill knight liability likelihood locked magic master merchant monk murder mystery newspaper night overtake ...*

piano*: accompanied accordion acoustic allegro alto arrangements artist ballad ballet band banjo bar baritone bass bassoon beginning bench brass brothers cello chamber choir choral chords chorus clarinet classic composed composition concert conduct conservatory dance disc dive drum duet duo ear electric ensemble evening fiddle fireplace flat flute folk forte grand guild guitar guy hammer happy harmonic harmony harp harper harpsichord hobbies horn improvisation inspired instrumentation instruments jazz key keyboard lesson listen lounge lyrics mandolin melody mezzo minor minority mood music musicians nocturnal oboe occasion opera opus orchestra organ overture pedagogy pedal percussion ...*

Figure 9: The hundred words with the highest mutual information with 'thief' and 'piano' on the WWW, in alphabetic order. These associations might be useful for OCR, or speech recognition.

of words, discovered automatically from WWW text, give an idea of what is associated with the given word. Such information might prove useful in deciding between possible readings of words in speech recognition or in optical character recognition.

5.3 Dependency relations and a sample entry

We have mentioned that NLP parsing tools can extract dependency relations between words. We think that the Very Large Lexicon should have (in addition to entries for individual words containing frequency, grammatical, normalization and collocation information) entries for each dependency relation. Associated with each normalized dependency relation, is a frequency within the documents treated. There is also a link to other forms of the dependency relation involving other derivational forms of the words and other syntactic relations between them (e.g. between "...presidential election..." and "...elect...president...").

An example of such an entry is given in Figure 10. In this example, we imagine that the Very Large Lexicon is derived from domain classified documents (Doyle 1965, Chakrabarti, Dom, Agrawal and Raghavan 1998). In the example, the entire lexicon has been derived from documents categorized as "political." In addition, to variant forms of the dependency relation, the entry should contain both common words and other dependency relations found within some window around instance of the entry. Any recognized entities (Donaldson 1993) are also to be stored. The last part of the example, though other items might be included in the entry, is a set of pointers to dependency relations involving the words in the entry. These

LEXICON:	Politics	
ENTRY:	ADJ(presidential election)	
FREQUENCY:	27,486/100,000,000	
VARIANTS:	DOBJ(elect president)	SUBJPASS (president was elected)
	NNPREP(election of president)	NPDOBJ(elected president)
CONTEXT:	*50 words (frequency $>$ 5) before/after*	
	other entries found more than once in window, e.g. NN(acceptance speech)	
ENTITIES:	*other recognized people, places, things*	
NETWORK:	*pointers to lexical class members*	
	ADJ(presidential, *)	presidential things
	ADJ(*, election)	types of elections

Figure 10: Sample entry. An entry in the Very Large Lexicon for politics.

other entries form a sort of automatically generatable network (Grefenstette 1997) of concepts related to the entry.

6 Example of using a Very Large Lexicon

We can simulate the presence of a Very Large Lexicon containing information sketched in the previous section by using existing web browsers. Consider the following task, you must translate “groupe de travail” from French to English. The dictionary gives the following translations for “groupe”: *cluster, group, grouping, concern, collective.* For “travail” we have *work, labor, labour*. Suppose, in addition, that we know that French structures such as *A de B* are likely to be translated as *B A* in English. In an already constructed Very Large Lexicon you could just look up the most likely combination. In the meantime, we can use a web portal, such as Altavista. If we look up all the possible combinations and note their frequencies, we get: *labour cluster: 2; labor cluster: 6; labor grouping: 7; labour grouping: 17; work grouping: 31; work cluster: 107; labour group: 439; labor group: 724; work group: 66593*. Here the most common combination gives the best translation.

In (*The world wide web as a resource for example-based machine translation tasks* 1999) we tested this method on an entire German-English and Spanish-English dictionary. For all ambiguous compositional translations in both language pairs, the most frequent combinations on Web pages gives the right translations 86% and 87% of the time. Figure 11 and 12 show some examples of the generated ambiguous translation candidates and their frequencies on the Web.

This type of question, here lexical choice in translation, could be answered by a Very Large Lexicon. The same type of problems arises in other language applications such speech recognition and optical character recognition. Many other NLP applications could benefit from having a large abstracted model what structures and word combinations are commonly used, and with what frequencies.

		Altavista	** if*	*MAX if*
compound	*candidate*	*frequency*	*gold standard*	*most frequent*
Angebotspreis	offer price	9767	*	MAX
Angebotspreis	offer prize	206	-	
Apfelkraut	apple herb	167	-	MAX
Apfelkraut	apple syrup	159	*	
Apfelsaft	apple juice	13841	*	MAX
Apfelsaft	apple sap	25	-	
Appartementhaus	apartment chop	0	-	
Appartementhaus	apartment cut	127	-	
Appartementhaus	apartment house	8356	*	MAX
Appartementhaus	apartment rampage	0	-	
Appartementhaus	flat chop	10	-	
Appartementhaus	flat cut	621	-	
Appartementhaus	flat house	882	-	
Appartementhaus	flat rampage	0	-	
Bogenbrücke	arch bridge	2304	*	MAX
Bogenbrücke	bow bridge	224	-	

Figure 11: Ambiguous German term translations, using the translations of parts of compounds words The Altavista count corresponds to the number of times the English candidate was found there and the next two columns show whether the given English translation was the one given by the dictionary for the entire compound, and whether it was the most frequent on the Web. 86* and MAX in the last two columns, showing that the most common combination was the translation given by the dictionary.

compound	*candidate*	*Altavista frequency*	** if gold standard*	*MAX if most frequent*
agregado-de-prensa	press-attaché	403	*	MAX
agregado-de-prensa	squeezer-attaché	0	-	
agua-corriente	common-water	2815	-	
agua-corriente	current-water	5213	-	
agua-corriente	draft-water	1438	-	
agua-corriente	draught-water	11	-	
agua-corriente	flowing-water	13264	-	
agua-corriente	going-water	343	-	
agua-corriente	ordinary-water	2040	-	
agua-corriente	power-water	12695	-	
agua-corriente	running-water	49358	*	MAX
agua-corriente	stream-water	9264	-	
agua-corriente	usual-water	1252	-	
agua-mineral	mineral-water	33058	*	MAX
agua-mineral	ore-water	178	-	
agua-salada	pickle-water	284	-	
agua-salada	salt-water	98690	*	MAX
águila-real	actual-eagle	60	-	
águila-real	essential-eagle	11	-	
águila-real	real-eagle	176	-	
águila-real	royal-eagle	431	*	MAX
ahorro-de-energa	decisiveness-saving	0	-	
ahorro-de-energa	energy-saving	140148	*	MAX

Figure 12: Ambiguous Spanish term translations, using the translations of parts of compounds words. The Altavista count corresponds to the number of times the English candidate was found there and the next two columns show whether the given English translation was the one given by the dictionary for the entire compound, and whether it was the most frequent on the Web. 87and MAX in the last two columns, showing that the most common combination was the translation given by the dictionary.

7 Conclusion

As a summary of this chapter, we have said that computers are useful, but only when we can reduce the problems we want to treat to their level. Language Models allow us to remove detail from text and make different things look similar so that a computer can treat them. The World Wide Web presents us now with a tremendous amount of text from which we can extract models of how language is used. Current Natural Language Processing tools can deal with the large amounts of text that the Web provides. With these tools, we can classify texts and extract abstracted models of how words interact. We can store these large models in a new structure, Very Large Lexicons. These models are huge but computer memory is cheap and becoming cheaper. With such Very Large Lexicons, automatically extracted from the WWW, we can solve many natural language processing problems, and imagine newer and more powerful natural language processing applications.

References

Abney, S.(1991), Parsing by chunks, *in* S. A. Robert Berwick and C. Tenny (eds), *Principle-Based Parsing*, Kluwer Academic Publishers, Dordrecht.

Ait-Mokhtar, S. and Chanod, J.-P.(1997), Incremental finite-state parsing, *ANLP'97*, Washington, pp. 72–79.

Chakrabarti, S., Dom, B., Agrawal, R. and Raghavan, P.(1998), Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies, *VLDB Journal: Very Large Data Bases* **7**(3), 163–178.

Chanod, J. and Tapanainen, P.(1995), Creating a tagset, lexicon and guesser for a french tagger, *Proceedings of the ACL SIGDAT Workshop*, Dublin, Ireland.

Church, K.(1988), A stochastic parts program and noun phrase parser for unrestricted text, *Proceedings of the 2nd Conference on Applied Natural Language Processing* pp. 136–143.

Church, K. W. and Hanks, P.(1990), Word association norms, mutual information, and lexicography, *Computational Linguistics* **16**(1), 22–29.

Debili, F.(1982), *Analyse Syntaxico-Semantique Fondee sur une Acquisition Automatique de Relations Lexicales-Semantiques*, PhD thesis, University of Paris XI, France.

Donaldson, D. D.(1993), Internal and external evidence in the identification and semantic categorization of proper names, *in* B. Boguraev and J. Pustejovsky (eds), *Proceedings of the SIGLEX Workshop on Acquisition of Lexical Knowledge from Text*, Columbus, OH, pp. 32–43.

Doyle, L. B.(1965), Is automatic classification a reasonable application of statistical analysis of text?, *Journal of the ACM* **12**(4), 473–489.

Ejerhed, E. and Church, K.(1983), Finite state parsing, *in* F. Karlsson (ed.), *Papers from the Seventh Scandinavian Conference of Linguistics*, University of Helsinki, Department of General Linguistics, pp. 410–432.

Frakes, W. B. and Baeza-Yates, R. (eds)(1992), *Information Retrieval: Data Structures and Algorithms*, Prentice Hall, New Jersey.

Ghani, R., Jones, R., Mladenic, D., Nigam, K. and Slattery, S.(n.d.), Data mining on symbolic knowledge extracted from the web, *Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining*.

Grefenstette, G.(1997), Sqlet : Short query linguistic expansion techniques: Palliating one or two-word queries by providing intermediate structure to text, *RIAO'97, Computer-Assisted Information Searching on the Internet*, Montreal, Canada.

Grefenstette, G.(1999), Light parsing as finite-state filtering, *in* A. Kornai (ed.), *Extended Finite State Models of Language*, number 0-521-63198-x, Cambridge University Press.

Grefenstette, G. and Nioche, J.(2000), Estimation of english and non-english language use on the www, *Proceedings of RIAO'2000, Content-Based Multimedia Information Access*, Paris, pp. 237–246. http://arXiv.org/find/cs/1/au:+nioche/0/1/0/past/0/1.

Hindle, D.(1993), A parser for text corpora, *in* B. Atkins and A. Zampolli (eds), *Computational Approaches to the Lexicon*, Clarendon Press.

Hutchins, W. J. and Somers, H. L.(1992), *An Introduction to Machine Translation*, Academic Press, New York.

Jacquemin, C.(1999), Syntagmatic and paradigmatic representations of term variation, *Proc. of 37th Annual Meeting of the Association for Computational Linguistics*.

Jensen, K., Heidorn, G. E. and Richardson, S. D. (eds)(1993), *Natural Language Processing: The PLNLP Approach*.

Karttunen, L., Chanod, J., Grefenstette, G. and Schiller, A.(1996), Regular expression for language engineering, *Natural Language Engineering*.

Lawrence, S. and Giles, C. L.(1999), Accessibility of information on the web, *Nature* **400**, 107–109.

Nievergelt, J., Gasser, R., Maeser, F. and Wirth, C.(1995), All the needles in a haystack: Can exhaustive search overcome combinatorial chaos?, *Lecture Notes in Computer Science* **1000**, 254–276.

Pavia, N. G.(2000), Heterogeneite des corpus: vers un parseur robuste reconfigurable et adaptable, *RECITAL (student session of TALN conference)*.

Pereira, F. C. N. and Warren, D. H. D.(1980), Definite clause grammars for language analysis—A survey of the formalism and a comparison with augmented transition networks, *Artificial Intelligence* **13**(3), 231–278.

Placeway, P., Schwartz, R., Fung, P. and Nguyen, L.(1993), The estimation of powerful language models from small and large corpora, *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing II*, pp. 33–36.

Porter, M. F.(1980), An algorithm for suffix stripping, *Program* **14**(3), 130–137.

Robert Gaizauskas, A. M. R.(1997), Coupling information retrieval and informatin extraction: A new text technology for gathering information from the web, *RIAO'97, Computer-Assisted Information Searching on the Internet*, Montreal, Canada, pp. 356–376.

Schaller, R.(1997), Moore's law: past, present and future, *IEEE Spectrum*.

The world wide web as a resource for example-based machine translation tasks(1999).

Tymoczko, T.(1990), The four-color problem and its philosophical significance, *in* J. L. Garfield (ed.), *Foundations of cognitive science: the essential readings*, Paragon House.

Voutilainen, A., Heikkila, J. and Anttila, A.(1992), A lexicon and constraint grammar of english, *Proceedings of the Fourteenth International Conference on Computational Linguistics*, COLING'92, Nantes, France.

Yang, Y. and Pedersen, J. O.(1997), A comparative study on feature selection in text categorization, *Proc. 14th International Conference on Machine Learning*, Morgan Kaufmann, pp. 412–420.