On the Instability of Using Search Engine Page Hits as a Proxy for n-gram Frequencies

Preslav Nakov and **Marti Hearst**
EECS and SIMS
University of California at Berkeley
Berkeley, CA 94720
nakov@cs.berkeley.edu, hearst@sims.berkeley.edu

Abstract

The idea of using the Web as a corpus for linguistic research is getting increasingly popular. Most often this means using page hit counts as an estimate for n-gram frequencies. While the results so far have been very encouraging, there are also some problems, the most important of which is the instability of these estimates. Using a particular NLP task, we find substantial variability in the n-gram counts across different search engines as well as for the same search engine across time. In addition, we show the potential of taking advantage of these variabilities by combining estimates from different search engines.

1 Introduction

In 2001, (Banko & Brill 01) advocated for the creative use of very large text collections as an alternative to sophisticated algorithms and hand-built resources. They demonstrated the idea on a lexical disambiguation problem for which labeled examples are available "for free". The problem was to choose which of 2-3 commonly confused words (e.g., {*principle, principal*}) were appropriate for a given context. The labeled data was "free" because the authors could safely assume that in the carefully edited text in their training set the words are used correctly. They show that even using a very simple algorithm, the results continue to improve log-linearly with more training data, even out to a billion words. They conclude that getting more data may work better idea than fine-tuning algorithms. Today, the obvious source of very large data is the Web.

Using the Web as a training and testing corpus is attracting ever-increasing attention. In 2003 the journal *Computational Linguistics* had a special issue (Kilgariff & Grefenstette 03), and in 2005 the Corpus Linguistics conference includes a special workshop on the Web as Corpus. The Web has been used as a corpus for a variety of NLP tasks including, but not limited to: machine translation (Grefenstette 98; Resnik 99; Cao & Li 02; Way & Gough 03), question answering: (Dumais *et al.* 02; Soricut & Brill 04), word sense disambiguation (Mihalcea & Moldovan 99; Rigau *et al.* 02; Santamaría *et al.* 03; Zahariev 04), extraction of semantic relations, (Chklovski & Pantel 04; Idan Szpektor & Coppola 04; Shinzato & Torisawa 04), anaphora resolution: (Modjeska *et al.* 03), prepositional phrase attachment: (Volk 01; Calvo & Gelbukh 03), language modeling: (Zhu & Rosenfeld 01; Keller & Lapata 03), on so on.

Despite the variability of applications, the most popular use of the Web as a corpus is as a means to obtain page hit counts as an estimate for n-gram word frequencies. (Keller & Lapata 03) demonstrate a high correlation between page hits and corpus bigram frequencies, as well as between page hits and plausibility judgments. They propose using Web counts as a baseline unsupervised method for many NLP tasks and later experimented with eight NLP problems (machine translation candidate selection, spelling correction, adjective ordering, article generation, noun compound bracketing, noun compound interpretation, countability detection and prepositional phrase attachment), and show that variations on n-gram counts often perform nearly as well as more elaborate methods (Lapata & Keller 05). More recently, we have shown that the Web has the potential for more than just a baseline. Using various Web-derived surface features, in addition to paraphrases and n-gram counts which go beyond the standard $n = 1, 2, 3$, we demonstrated state-of-the-art results on the task of noun compound bracketing (Nakov & Hearst 05).

2 Problems and Limitations

Web search engines provide a convenient way for researchers to obtain statistics over an enormous corpus, but using search engines for this purpose is not without drawbacks.

First, the use of Web search engines imposes limitations on what kinds of queries can be is-

sued, mainly because of the lack of linguistic annotation. For example, if we want to estimate the probability that *health* precedes *care*: $\frac{\#(''health\ care'')}{\#(care)}$, we need the frequencies of *"health care"* and *care*, where both words are nouns. The problem is that a query for *care* will return many pages where it is used as a verb, while *health care* it would nearly always occur as a noun. Even when both *health* and *care* are used as nouns and are adjacent, they may belong to different NPs but sit next to each other only by chance. Furthermore, since search engines ignore punctuation characters, the two nouns may also come from different sentences.

Other Web search engine restrictions prevent querying directly for terms containing hyphens or possessive markers such as *amino-acid sequence* and *protein synthesis' inhibition*. They also disallow querying for a term like *bronchoalveolar lavage (BAL) fluid*, which contains an internal parenthesized abbreviation. Further, search engines cannot support queries that make use of generalized POS information such as

stem cells VERB PREP DET brain

in which the uppercase patterns stand for any verb, any preposition and any determiner, e.g., *stem cells derived from the brain*.

Furthermore, using page hits as a proxy for n-gram frequencies can produce some counterintuitive results. Consider the bigrams w_1w_4, w_2w_4 and w_3w_4 and a page that contains each bigram exactly once. A search engine will contribute a page count of 1 for w_4 instead of a frequency of 3; thus the page hits for w_4 can be smaller than the page hits for the sum of the individual bigrams. See (Keller & Lapata 03) for more potential problems with page hits.

Another important problem is instability of the n-gram counts. Today Web search engines are too complex to be run on a single machine, and instead the queries are served by hundreds, sometimes thousands of servers, which collaborate to produce the final result. In addition, the Web is dynamic, since at any given time some pages disappear, some appear for the first time, and some are frequently updated. Thus search engines need to update their indexes frequently, and in fact the different engines compete on how "fresh" their indexes are. Thus, the number of page hits for a given query changes over time in unpredictable ways.

The indexes themselves are too big to be stored on a single machine and so are spread across multiple machines (Brin & Page 98). For availability and efficiency reasons, there are also multiple copies of the same part of the index, and these are often not in always in synch with one another since the different copies are updated at different times. As a result, if we issue the same query multiple times in rapid succession, we may connect to different physical machines and get different results. This is known as search engine "dancing".

With a little bit of luck, one can observe Google "dancing" by comparing the results of different data centers, e.g., `www.google.com`, `www2.google.com`, `www3.google.com`. Alternatively, one can try the *Google dance tool* at: `http://www.seochat.com/googledance`. More on the phenomenon can be found on the Web, e.g., at `http://dance.efactory.de/` or by querying some search engine for the exact phrase "Google dancing".

From a research perspective, "dancing" and dynamics over time are undesirable, as they preclude the exact replicability of any results obtained using search engines. At best, one could reproduce the same initial conditions, and expect similar outcomes.

Another undesirable aspect of using Web search engines is that two of the major engines (Google and Yahoo) do not provide exact numbers of page hits, but instead show rounded estimates. For example, at the moment of preparation of this paper, Google returns 79,000,000 page hits as a result of a query for the exact phrase *"search engine"*, and Yahoo Search returns 127,000,000. Google and Yahoo provide exact numbers of page hits only in case this number is relatively small. MSN Search, by contrast, does not round its page hits, and for the "search engines" query it returns 46,502,549.

This rounding is most likely done because for most users' purposes, exact counts are not necessary once the numbers get somewhat large, and computing the exact numbers is expensive if the index is distributed and continually changing.

It is unclear what the implications of these inconsistencies are on using the Web to obtain n-gram frequencies. If the estimates are close to accurate and consistent across queries, this should not have a big impact for most applications, since most of these only need the ratios of different n-

grams. Probably a bigger problem is posed by the rounding of the estimates.

We decided that the best way to determine the impact of rounding and inconsistencies was to design a suit of experiments organized around a real NLP task. We chose noun compound bracketing, which, while being a simple task, can be solved using several different methods which make use of n-grams of different lengths. In the next two sections we first describe the noun compound bracketing problem, and then describe comparative experiments on this problem.

3 Noun Compound Bracketing

Consider the following contrastive pair of noun compounds:

(1) *liver cell antibody*
(2) *liver cell line*

In example (1) an *antibody* targets a *liver cell*, while (2) refers to a *cell line* which is derived from the *liver*. Although equivalent at the part of speech (POS) level, these two noun compounds have different syntactic trees. The distinction can be represented as a binary tree or, equivalently, as a binary bracketing:

(1b) [[*liver cell*] *antibody*] (left bracketing)
(2b) [*liver* [*cell line*]] (right bracketing)

3.1 Unigrams and Bigrams

The problem of choosing the correct bracketing has been traditionally addressed using unigram and bigram frequencies (Marcus 80; Pustejovsky *et al.* 93; Resnik 93; Lauer 95; Lapata & Keller 05). In related work, a distinction is often made between what is called the *dependency model* and the *adjacency model* (Lauer 95). The main idea is as follows. For a given 3-word NC $w_1w_2w_3$, there are two reasons it may take on right bracketing, $[w_1[w_2w_3]]$. Either (a) w_2w_3 is a compound (modified by w_1), or (b) w_1 and w_2 independently modify w_3. This distinction can be seen in the examples *home health care* (*health care* is a compound modified by *home*) versus *adult male rat* (*adult* and *male* independently modify *rat*).

The adjacency model checks (a), whether w_2w_3 is a compound (i.e., how strongly w_2 modifies w_3 as opposed to w_1w_2 being a compound) to decide whether or not to predict a right bracketing. The dependency model checks (b), does w_1 modify w_3 (as opposed to w_1 modifying w_2).

Adjacency and dependency could be computed via frequencies, but we can also use probabilities. Let $\Pr(w_i \rightarrow w_j|w_j)$ be the probability that the word w_i precedes a given fixed word w_j. So in a dependency model we can compare $\Pr(w_1 \rightarrow w_3|w_3)$ to $\Pr(w_1 \rightarrow w_2|w_2)$. The alternative adjacency model compares $\Pr(w_2 \rightarrow w_3|w_3)$ to $\Pr(w_1 \rightarrow w_2|w_2)$, i.e., the association strength between the last two words vs. that between the first two. If the first probability is larger than the second, the model predicts right.

The probability $\Pr(w_1 \rightarrow w_2|w_2)$ can be estimated as $\#(w_1, w_2)/\#(w_2)$, where $\#(w_1, w_2)$ and $\#(w_2)$ are the corresponding bigram and unigram frequencies. They can be approximated as the number of pages returned by a search engine in response to queries for the exact phrase "w_1 w_2" and for the word w_2. In our experiments below we smoothed[1] each of the frequencies by adding 0.5 to avoid problems caused by nonexistent n-grams.

In both models, $\Pr(w_i \rightarrow w_j|w_j)$ can be replaced by some (possibly symmetric) measure of association between w_i and w_j (Nakov & Hearst 05). Below we use *Chi squared* (χ^2) and mutual information (MI). See (Nakov & Hearst 05) for details on how to compute χ^2.

3.2 Longer n-grams

As the Web is a very big corpus, we can get reliable estimates for longer n-grams too. Below we list some other kinds of statistics that can be computed from the Web corpus that we have found helpful in other work (Nakov & Hearst 05), and that are used in the experiments in the next section.

First, the genitive ending, or *possessive* marker, can be a useful indicator. The phrase *brain's stem cells* suggests a right bracketing for *brain stem cells*, while *brain stem's cells* favors a left bracketing. In some cases, we can query for this directly: although search engines drop the apostrophe, they keep the *s*, so we can query for *"brain's"* (but not for *"brains' "*). We then compare the number of times the possessive marker appeared on the second versus the first word, to make a bracketing decision.

Abbreviations are another important feature. For example, *"tumor necrosis factor (NF)"* suggests a right bracketing, while *"tumor necrosis*

[1] Zero counts sometimes happen for $\#(w_1, w_3)$, but are rare for unigrams and bigrams on the Web, and there is no need for a more sophisticated smoothing.

(TN) factor" would favor left. We would like to issue exact phrase queries for the two potential abbreviation patterns and see which one is more frequent. Unfortunately, the search engines drop the brackets and ignore the capitalization, so we issue queries with the parentheses removed, as in *"tumor necrosis factor nf"*. This produces highly accurate results, although errors occur when the abbreviation is an existing word (e.g., *me*), a state (e.g., *CA*), a Roman digit (e.g., *IV*), etc.

Another reliable feature is *concatenation*. Consider the NC *health care reform*, which is left-bracketed. Now, consider the bigram *"health care"*. At the time of writing, Google estimates 80,900,000 pages for it as an exact term. Now, if we try the word *healthcare* we get 80,500,000 hits. At the same time, *carereform* returns just 109. This suggests that authors sometimes concatenate words that act as compounds. We find below that comparing the frequency of the concatenation of the left bigram to that of the right (adjacency model for concatenations) often yields accurate results. We also tried the dependency model for concatenations, as well as the concatenations of two words in the context of the third one (i.e., compare frequencies of *"healthcare reform"* and *"health carereform"*).

Further, we try to look inside the *internal inflection variability*. The idea is that if *"tyrosine kinase activation"* is left-bracketed, then the first two words probably make a whole and thus the second word can be found inflected elsewhere but the first word cannot, e.g., *"tyrosine kinase**s** activation"*. Alternatively, if we find different internal inflections of the first word, this would favor a right bracketing.

Finally, we try switching the word order of the first two words. If they independently modify the third one (which implies a right bracketing), then we could expect to see also a form with the first two words switched, e.g., if we are given *"adult male rat"*, we would also expect *"male adult rat"*.

4 Experiments and Results

We experimented with the dataset from (Lauer 95), in order to produce results comparable to those of Lauer and Keller & Lapata. The set consists of 244 unambiguous 3-word noun compounds extracted from *Grolier's encyclopedia*; however, only 216 of these NCs are unique.

(Lauer 95) derived n-gram frequencies from the

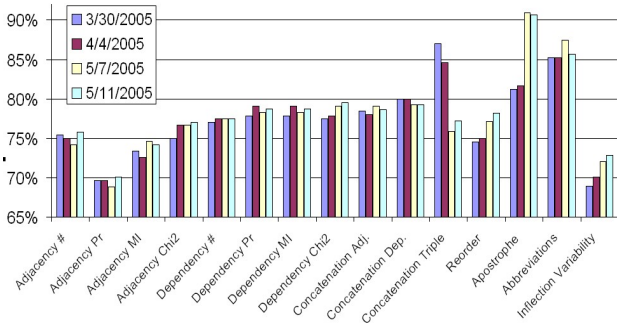

Figure 1: **Comparison over time for Google.** Precision for any language, no inflections.

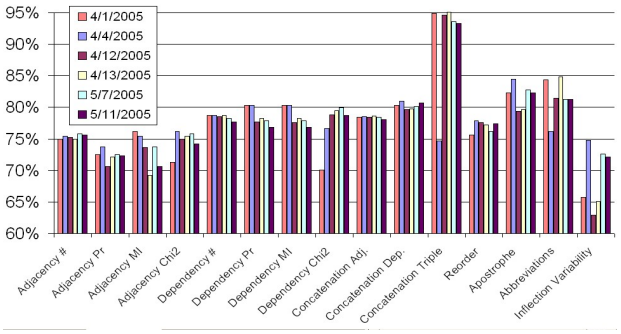

Figure 2: **Comparison over time for MSN Search.** Precision for any language, no inflections.

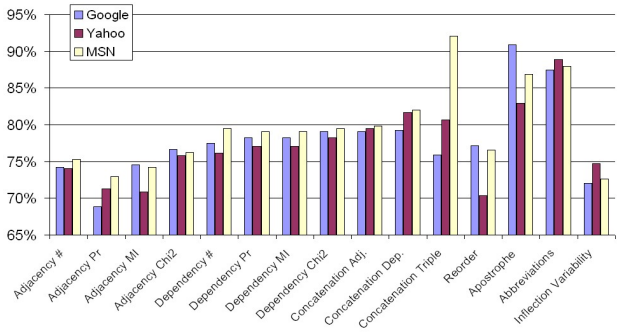

Figure 3: **Comparison by search engine.** *Precision* (in %) for any language, no inflections. All results are for 5/7/2005.

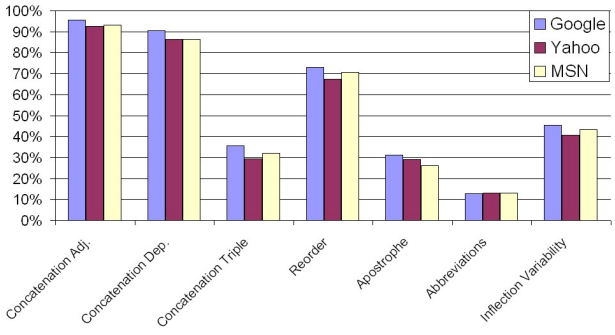

Figure 4: **Comparison by search engine.** *Recall* (in %) for any language, no inflections. All results are for 5/7/2005.

Grolier's corpus and tested the dependency and the adjacency models using this text. To help combat data sparseness issues he also incorporated a taxonomy and some additional information (see Related Work section above).

We performed a series of experiments comparing the accuracy of the methods described above across four dimensions: search engine (Google vs. Yahoo vs. MSN), time, language filter usage (English or no filter) and inflected wordforms usage.

At the time of writing, the Google search engine reportedly indexes more than 8 billion pages, i.e., about 8 trillion words, which is about 80,000 times the size of the British National Corpus (100 million words), thus confirming it as a gateway to a very large corpus. We were unable to find official information about the sizes of Yahoo and MSN Search, but they probably index a similar number of pages. When still in Beta version, MSN announced indexing over 5 billion pages.

For all n-grams, we issued exact phrase queries within a single day. Unless otherwise stated, the queries were not inflected and no language filter was applied. We used a threshold of 5 for the difference between the left- and the right-predicting n-gram frequencies: we did not make a decision when the module of that difference was below the threshold. This slightly lowers the recall but potentially increases the precision.

Figures 1 and 2 show the variability over time for Google and for MSN respectively. (As Yahoo behaves similarly to Google, it is omitted here due to space limitations.) We chose time samples at varying time intervals in an attempt to capture index changes, n case they happen in the same fixed time intervals. For Google (see 1), we observe a low variability in the adjacency- and dependency-based models and a more sizable variability for the other models and features. The variability is especially high for *apostrophe* and *concatenation triple*: while in the first two time snapshots the precision of the apostrophes is much lower than in the last two, it is the reverse for concatenation.

MSN exhibits a more uniform behavior overall (see 2), however while the variability in the adjacency- and dependency-based models is still a little bit lower than that of the last five features, it is more significant than to Google's. We think that this is due to the rounding: because Google's counts are rounded, they change less over time, especially for very large counts. By contrast, these counts are exact for MSN search, which makes its unigram and bigram counts more sensitive to variation. For the higher order n-grams, both engines exhibit a higher variability: these counts are smaller, and so are more likely to be represented by exact numbers in Google, and thus they are also more sensitive to index updates for both search engines.

Figure 3 compares the three search engines at the same fixed time point: 5/7/2005. The biggest difference in precision is exhibited by *concatenation triple* which in MSN search achieves a precision of 92%, which is better than the others' by 11%. Other large variations are seen in *apostrophe*, *reorder*, and to a lesser extent in the adjacency- and dependency-based models. As we expected, MSN Search looks best overall (especially on the unigram- and bigram-based models), which we attribute to the better accuracy of its n-gram estimates. Google is almost 5% ahead of the others on *apostrophes* and *reorder*. Yahoo leads on *abbreviations* and *inflection variability*. The fact that different search engines exhibit strength on different kinds of queries and models shows the potential of combining them: in a majority vote combining some of the best models, we would choose *concatenation triple* from MSN Search and *apostrophe* from Google and *abbreviations* from Yahoo (together with *concatenation dependency*, χ^2 *dependency* and χ^2 *adjacency*). Figure 4 shows the corresponding recall for some of the methods (the recall is always 100% for the rest). We can see that Google consistently exhibits a slightly higher recall, which suggests it might have a bigger index compared to Yahoo and MSN Search.

Figure 5 compares, on a fixed date (5/7/2005),

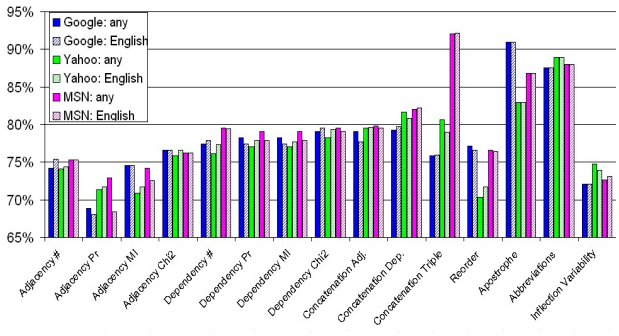

Figure 5: **Comparison by search engine: any language vs. English.** *Precision* shown in %, no inflections. All results are for 5/7/2005.

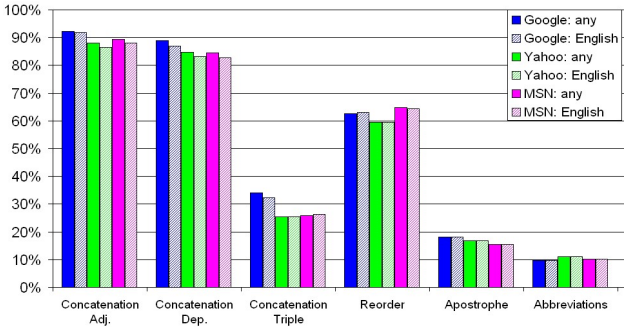

Figure 6: **Comparison by search engine: any language vs. English.** *Recall* shown in %, no inflections. All results are for 5/7/2005.

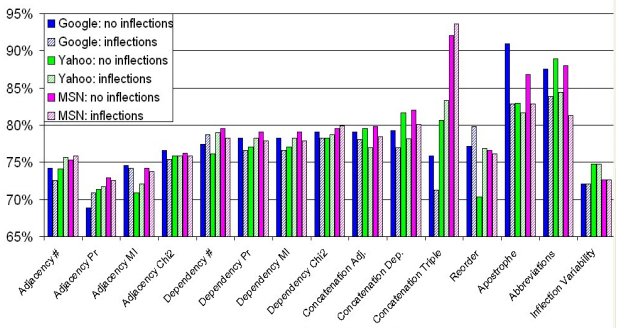

Figure 7: **Comparison by search engine: no inflections vs. using inflections.** *Precision* shown in %, any language. All results are for 5/7/2005.

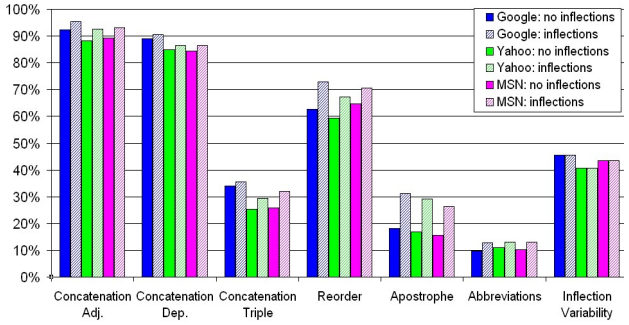

Figure 8: **Comparison by search engine: no inflections vs. using inflections.** *Recall* shown in %, any language. All results are for 5/7/2005.

for all the three search engines the impact of language filtering, meaning requiring only documents in English versus no restriction on language. The impact of the language filter on the precision seems minor and inconsistent for all three search engines: sometimes the results are improved slightly and sometimes they are negatively impacted. Figure 6 compares the corresponding recall for some of the models (the rest are omitted as the recall for them is 100%). As we can see, using English only leads to a drop in recall, as one could expect, but this drop is small.

Finally, Figure 7 compares for the three search engines the impact of using inflections[2]. When we estimate the frequency of a word, e.g., *tumor*, we also add up the frequencies of all possible variants, e.g., *tumors*, *tumour*, *tumours*. For bigrams, we inflect only the second word, and for n-grams only the last one. The results are again mixed, but the impact on precision is more significant compared to that of the language filter, especially on the high-order n-grams (of course, there is no impact on *inflection variability*). Figure 8 compares the corresponding recall for some of the models (the rest are omitted as they are 100%). As one would expect, the recall goes up when using inflection. The change for *apostrophe*, *reorder* and *concatenation triple* is again the biggest.

5 Conclusions and Future Work

Using a real NLP task, we have shown that effects of variability over time and across search engines, as well as using language filters and morphologically inflected wordforms can substantially effect

[2]We made use of Carroll's morphological tools: http://www.cogs.susx.ac.uk/lab/nlp/carroll/morph.html.

the results of an NLP application, and thus is important to keep in mind when interpreting results obtained using Web-derived n-gram frequencies.

In order to further bolster these results we will need to perform similar studies for other NLP tasks, which make use of Web-derived n-gram estimates. We would also like to run similar experiments for languages other than English, where the language filter could be much more important, and where the impact of the inflection variability would be differ, especially in case of a morphologically richer language.

Acknowledgements This research was supported by NSF DBI-0317510, and a gift from Genentech.

References

(Banko & Brill 01) Michele Banko and Eric Brill. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of ACL*, 2001.

(Brin & Page 98) Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1-7):107–117, 1998.

(Calvo & Gelbukh 03) Hiram Calvo and Alexander Gelbukh. Improving prepositional phrase attachment disambiguation using the web as corpus. In *Progress in Pattern Recognition, Speech and Image Analysis: 8th Iberoamerican Congress on Pattern Recognition, CIARP 2003*, 2003.

(Cao & Li 02) Yunbo Cao and Hang Li. Base noun phrase translation using web data and the EM algorithm. In *COLING*, pages 127–133, 2002.

(Chklovski & Pantel 04) Timothy Chklovski and Patrick Pantel. VerbOcean: Mining the web for fine-grained semantic verb relations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 33–40, 2004.

(Dumais *et al.* 02) Susan Dumais, Michele Banko, Eric Brill, Jimmy Lin, and Andrew Ng. Web question answering: Is more always better? In *Proceedings of SIGIR*, pages 291–298, 2002.

(Grefenstette 98) Gregory Grefenstette. The world wide web as a resourcefor example-based machine translation tasks. In *Proceedings of the ASLIB Conference on Translating and the Computer.*, 1998.

(Idan Szpektor & Coppola 04) Ido Dagan Idan Szpektor, Hristo Tanev and Bonaventura Coppola. Scaling web-based acquisition of entailment relations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 41–48, 2004.

(Keller & Lapata 03) Frank Keller and Mirella Lapata. Using the Web to obtain frequencies for unseen bigrams. *Computational Linguistics*, 29:459–484, 2003.

(Kilgariff & Grefenstette 03) Adam Kilgariff and Gregory Grefenstette. Introduction to the special issue on the web as corpus. *Computational Linguistics*, 29(3):333–347, 2003.

(Lapata & Keller 05) Mirella Lapata and Frank Keller. Web-based models for natural language processing. *ACM Transactions on Speech and Language Processing*, 2005.

(Lauer 95) Mark Lauer. *Designing Statistical Language Learners: Experiments on Noun Compounds*. Unpublished PhD thesis, Department of Computing Macquarie University NSW 2109 Australia, 1995.

(Marcus 80) Mitchell Marcus. *A Theory of Syntactic Recognition for Natural Language*. MIT Press, 1980.

(Mihalcea & Moldovan 99) Rada Mihalcea and Dan Moldovan. A method for word sense disambiguation of unrestricted text. In *ACL*, pages 152–158, 1999.

(Modjeska *et al.* 03) Natalia Modjeska, Katja Markert, and Malvina Nissim. Using the web in machine learning for other-anaphora resolution. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 176–183, 2003.

(Nakov & Hearst 05) Preslav Nakov and Marti Hearst. Search engine statistics beyond the n-gram: Application to noun compound bracketing. In *Proceedings of CoNLL-2005, Ninth Conference on Computational Natural Language Learning*, 2005.

(Pustejovsky *et al.* 93) James Pustejovsky, Peter Anick, and Sabine Bergler. Lexical semantic techniques for corpus analysis. *Computational Linguistics*, 19(2):331–358, 1993.

(Resnik 93) Philip Resnik. *Selection and information: a class-based approach to lexical relationships*. Unpublished PhD thesis, University of Pennsylvania, UMI Order No. GAX94-13894, 1993.

(Resnik 99) Philip Resnik. Mining the web for bilingual text. pages 527–534, 1999.

(Rigau *et al.* 02) German Rigau, Bernardo Magnini, Eneko Agirre, and John Carroll. Meaning: A roadmap to knowledge technologies. In *Proceedings of COLING Workshop on A Roadmap for Computational Linguistics*, 2002.

(Santamaría *et al.* 03) Celina Santamaría, Julio Gonzalo, and Felisa Verdejo. Automatic association of web directories with word senses. *Computational Linguistics*, 29(3):485–502, 2003.

(Shinzato & Torisawa 04) Keiji Shinzato and Kentaro Torisawa. Acquiring hyponymy relations from web documents. In *Proceedings of HLT-NAACL*, pages 73–80, 2004.

(Soricut & Brill 04) Radu Soricut and Eric Brill. Automatic question answering: Beyond the factoid. In *Proceedings of HLT-NAACL*, pages 57–64, 2004.

(Volk 01) Martin Volk. Exploiting the WWW as a corpus to resolve PP attachment ambiguities. In *Proceedings of Corpus Linguistics*, 2001.

(Way & Gough 03) Andy Way and Nano Gough. wEBMT: developing and validating an example-based machine translation system using the world wide web. *Computational Linguistics*, 29(3):421–457, 2003.

(Zahariev 04) Manuel Zahariev. *A (Acronyms)*. Unpublished PhD thesis, School of Computing Science, Simon Fraser University, USA, 2004.

(Zhu & Rosenfeld 01) Xiaojin Zhu and Ronald Rosenfeld. Improving trigram language modeling with the world wide web. In *Proceedings of ICASSP*, pages I:533–536, 2001.