The Web as a Baseline: Evaluating the Performance of Unsupervised Web-based Models for a Range of NLP Tasks

Citation for published version:

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Publisher's PDF, also known as Version of record

Published In:

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
The Web as a Baseline: Evaluating the Performance of Unsupervised Web-based Models for a Range of NLP Tasks

Mirella Lapata
Department of Computer Science
University of Sheffield
211 Portobello St., Sheffield S1 4DP
mlap@dcs.shef.ac.uk

Frank Keller
School of Informatics
University of Edinburgh
2 Buccleuch Pl., Edinburgh EH8 9LW
keller@inf.ed.ac.uk

Abstract
Previous work demonstrated that web counts can be used to approximate bigram frequencies, and thus should be useful for a wide variety of NLP tasks. So far, only two generation tasks (candidate selection for machine translation and confusion-set disambiguation) have been tested using web-scale data sets. The present paper investigates if these results generalize to tasks covering both syntax and semantics, both generation and analysis, and a larger range of \( n \)-grams. For the majority of tasks, we find that simple, unsupervised models perform better when \( n \)-gram frequencies are obtained from the web rather than from a large corpus. However, in most cases, web-based models fail to outperform more sophisticated state-of-the-art models trained on small corpora. We argue that web-based models should therefore be used as a baseline for, rather than an alternative to, standard models.

1 Introduction
Keller and Lapata (2003) investigated the validity of web counts for a range of predicate-argument bigrams (verb-object, adjective-noun, and noun-noun bigrams). They presented a simple method for retrieving bigram counts from the web by querying a search engine and demonstrated that web counts (a) correlate with frequencies obtained from a carefully edited, balanced corpus such as the 100M words British National Corpus (BNC), (b) correlate with frequencies recreated using smoothing methods in the case of unseen bigrams, (c) reliably predict human plausibility judgments, and (d) yield state-of-the-art performance on pseudo-disambiguation tasks.

Keller and Lapata’s (2003) results suggest that web-based frequencies can be a viable alternative to bigram frequencies obtained from smaller corpora or recreated using smoothing. However, they do not demonstrate that realistic NLP tasks can benefit from web counts. In order to show this, web counts would have to be applied to a diverse range of NLP tasks, both syntactic and semantic, involving analysis (e.g., disambiguation) and generation (e.g., selection among competing outputs). Also, it remains to be shown that the web-based approach scales up to larger \( n \)-grams (e.g., trigrams), and to combinations of different parts of speech (Keller and Lapata 2003 only tested bigrams involving nouns, verbs, and adjectives). Another important question is whether web-based methods, which are by definition unsupervised, can be competitive alternatives to supervised approaches used for most tasks in the literature.

This paper aims to address these questions. We start by using web counts for two generation tasks for which the use of large data sets has shown promising results: (a) target language candidate selection for machine translation (Grefenstette, 1998) and (b) context sensitive spelling correction (Banko and Brill, 2001a,b). Then we investigate the generality of the web-based approach by applying it to a range of analysis and generations tasks, involving both syntactic and semantic knowledge: (c) ordering of prenominal adjectives, (d) compound noun bracketing, (e) compound noun interpretation, and (f) noun countability detection. Table 1 gives an overview of these tasks and their properties.

In all cases, we propose a simple, unsupervised \( n \)-gram based model whose parameters are estimated using web counts. We compare this model both against a baseline (same model, but parameters estimated on the BNC) and against state-of-the-art models from the literature, which are either supervised (i.e., use annotated training data) or unsupervised but rely on taxonomies to recreate missing counts.

<table>
<thead>
<tr>
<th>Task</th>
<th>( n )</th>
<th>POS</th>
<th>Ling</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT candidate select.</td>
<td>1,2</td>
<td>V, N</td>
<td>Sem</td>
<td>Generation</td>
</tr>
<tr>
<td>Spelling correction</td>
<td>1,2,3</td>
<td>Any</td>
<td>Syn/Sem</td>
<td>Generation</td>
</tr>
<tr>
<td>Adjective ordering</td>
<td>1,2</td>
<td>Adj</td>
<td>Sem</td>
<td>Generation</td>
</tr>
<tr>
<td>Compound bracketing</td>
<td>1,2</td>
<td>N</td>
<td>Syn</td>
<td>Analysis</td>
</tr>
<tr>
<td>Compound interpret.</td>
<td>1,2,3</td>
<td>N, P</td>
<td>Sem</td>
<td>Analysis</td>
</tr>
<tr>
<td>Countability detection</td>
<td>1,2</td>
<td>N, Det</td>
<td>Sem</td>
<td>Analysis</td>
</tr>
</tbody>
</table>

Table 1: Overview of the tasks investigated in this paper (\( n \): size of \( n \)-gram; POS: parts of speech; Ling: linguistic knowledge; Type: type of task)
2 Method

Following Keller and Lapata (2003), web counts for n-grams were obtained using a simple heuristic based on queries to the search engine Altavista.1 In this approach, the web count for a given n-gram is simply the number of hits (pages) returned by the search engine for the queries generated for this n-gram. Three different types of queries were used for the NLP tasks in the present paper:

Literal queries use the quoted n-gram directly as a search term for Altavista (e.g., the bigram "history changes" expands to the query "history changes").

Near queries use Altavista's NEAR operator to expand the n-gram; a NEAR b means that a has to occur in the same ten word window as b; the window is treated as a bag of words (e.g., "history changes" expands to "history" NEAR "changes").

Inflected queries are performed by expanding an n-gram into all its morphological forms. These forms are then submitted as literal queries, and the resulting hits are summed up (e.g., "history changes" expands to "history" NEAR "changes").

Following Keller and Lapata (2003), web counts for a given n-gram is simply the number of hits (pages) returned by the search engine for the queries generated for this n-gram. Three different types of queries were used for the NLP tasks in the present paper:

Literal queries use the quoted n-gram directly as a search term for Altavista (e.g., the bigram "history changes" expands to the query "history changes").

Near queries use Altavista's NEAR operator to expand the n-gram; a NEAR b means that a has to occur in the same ten word window as b; the window is treated as a bag of words (e.g., "history changes" expands to "history" NEAR "changes").

Inflected queries are performed by expanding an n-gram into all its morphological forms. These forms are then submitted as literal queries, and the resulting hits are summed up (e.g., "history changes" expands to "history" NEAR "changes").

For all of our tasks, we have to select either the best of several possible models or the best parameter setting for a single model. We therefore require a separate development set. This was achieved by using the gold standard data set from the literature for a given task and randomly dividing it into a development set and a test set (of equal size). We report the test set performance for all models for a given task, and indicate which model shows optimal performance on the development set (marked by a ‘#’ in all subsequent tables). We then compare the test set performance of this optimal model to the performance of the models reported in the literature. It is important to note that the figures taken from the literature were typically obtained on the whole gold standard data set, and hence may differ from the performance on our test set. We work on the assumption that such differences are negligible.

We use χ² tests to determine whether the performance of the best web model on the test set is significantly different from that of the best BNC model. We also determine whether both models differ significantly from the baseline and from the best model in the literature. A set of diacritics is used to indicate significance throughout this paper, see Table 2.

3 Candidate Selection for Machine Translation

Target word selection is a generation task that occurs in machine translation (MT). A word in a source language can often be translated into different words in the target language and the choice of the appropriate translation depends on a variety of semantic and pragmatic factors. The task is illustrated in (1) where there are five translation alternatives for the German noun Geschichte listed in curly brackets, the first being the correct one.

(1) a. Die Geschichte ändert sich, nicht jedoch die Geographie.

b. {History, story, tale, saga, strip} changes but geography does not.

Statistical approaches to target word selection rely on bilingual lexica to provide all possible translations of words in the source language. Once the set of translation candidates is generated, statistical information gathered from target language corpora is used to select the most appropriate alternative (Dagan and Itai, 1994). The task is somewhat simplified by Grefenstette (1998) and Prescher

---

1 We did not use Google counts, as Google limits the number of queries to 1000 per day, which makes the process of retrieving a large number of web counts very time consuming.

2 The tools can be downloaded from http://www.cogs.susx.ac.uk/lab/nlp/carroll/morph.html.
et al. (2000) who do not produce a translation of the entire sentence. Instead, they focus on specific syntactic relations. Grefenstette translates compounds from German and Spanish into English, and uses BNC frequencies as a filter for candidate translations. He observes that this approach suffers from an acute data sparseness problem and goes on to obtain counts for candidate compounds through web searches, thus achieving a translation accuracy of 86–87%.

Prescher et al. (2000) concentrate on verbs and their objects. Assuming that the target language translation of the verb is known, they select from the candidate translations the noun that is most semantically compatible with the verb. The semantic fit between a verb and its argument is modeled using a class-based lexicon that is derived from unlabeled data using the expectation maximization algorithm (verb-argument model). Prescher et al. also propose a refined version of this approach that only models the fit between a verb and its object (verb-object model), disregarding other arguments of the verb. The two models are trained on the BNC and evaluated against two corpora of 1,340 and 814 bilingual sentence pairs, with an average of 8.63 and 2.83 translations for the object noun, respectively. Table 4 lists Prescher et al.’s results for the two corpora and for both models together with a random baseline (select a target noun at random) and a frequency baseline (select the most frequent target noun).

Grefenstette’s (1998) evaluation was restricted to compounds that are listed in a dictionary. These compounds are presumably well-established and fairly frequent, which makes it easy to obtain reliable web frequencies. We wanted to test if the web-based approach extends from lexicalized compounds to productive syntactic units for which dictionary entries do not exist. We therefore performed our evaluation using Prescher et al.’s (2000) test set of verb-object pairs. Web counts were retrieved for all possible verb-object translations; the most retrieved for all possible verb-object translations; the most

The web counts were gathered using inflected queries involving the verb, a determiner, and the object (see Section 2). Table 3 compares the web-based models against the BNC models. For both the high ambiguity and the low ambiguity data set, we find that the performance of the best Altavista model is not significantly different from that of the best BNC model. Table 4 compares our simple, unsupervised methods with the two sophisticated class-based models discussed above. The results show that there is no significant difference in performance between the best model reported in the literature and the best Altavista or the best BNC model. However, both models significantly outperform the baseline. This holds for both the high and low ambiguity data sets.

<table>
<thead>
<tr>
<th>Model</th>
<th>Altavista high ambig</th>
<th>BNC high ambig</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>45.74</td>
<td>68.73</td>
</tr>
<tr>
<td></td>
<td>45.16†</td>
<td>68.73†</td>
</tr>
<tr>
<td>Prescher et al. (2000): verb-object</td>
<td>45.30†</td>
<td>61.50</td>
</tr>
<tr>
<td>Best Altavista</td>
<td>45.16†</td>
<td>68.73†</td>
</tr>
<tr>
<td>Best BNC</td>
<td>45.89†</td>
<td>70.06†</td>
</tr>
<tr>
<td>Prescher et al. (2000): verb-object</td>
<td>49.40†</td>
<td>68.20</td>
</tr>
</tbody>
</table>

Table 3: Performance of Altavista counts and BNC counts for candidate selection for MT (data from Prescher et al. 2000)

<table>
<thead>
<tr>
<th>Model</th>
<th>high ambig</th>
<th>low ambig</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random baseline</td>
<td>14.20</td>
<td>45.90</td>
</tr>
<tr>
<td>Frequency baseline</td>
<td>31.90</td>
<td>45.50</td>
</tr>
<tr>
<td>Prescher et al. (2000): verb-argument</td>
<td>43.30</td>
<td>61.50</td>
</tr>
<tr>
<td>Best Altavista</td>
<td>45.16†</td>
<td>68.73†</td>
</tr>
<tr>
<td>Best BNC</td>
<td>45.89†</td>
<td>70.06†</td>
</tr>
<tr>
<td>Prescher et al. (2000): verb-object</td>
<td>49.40†</td>
<td>68.20</td>
</tr>
</tbody>
</table>

Table 4: Performance comparison with the literature for candidate selection for MT

4 Context-sensitive Spelling Correction

Context-sensitive spelling correction is the task of correcting spelling errors that result in valid words. Such a spelling error is illustrated in (4) where principal was typed when principle was intended.

(2) Introduction of the dialogue principal proved strikingly effective.

The task can be viewed as generation task, as it consists of choosing between alternative surface realizations of a word. This choice is typically modeled by confusion sets such as {principal, principle} or {then, than} under the assumption that each word in the set could be mistakenly typed when another word in the set was intended. The task is to infer which word in a confusion set is the correct one in a given context. This choice can be either syntactic (as for {then, than}) or semantic (as for {principal, principle}).

A number of machine learning methods have been proposed for context-sensitive spelling correction. These include a variety of Bayesian classifiers (Golding, 1995; Golding and Schabes, 1996), decision lists (Golding, 1995) transformation-based learning (Mangu and Brill, 1997), Latent Semantic Analysis (LSA) (Jones and Martin, 1997), multiplicative weight update algorithms (Golding and Roth, 1999), and augmented mixture models (Cucerzan and Yarowsky, 2002). Despite their differences, most approaches use two types of features: context words and collocations. Context word features record the presence of a word within a fixed window around the target word (bag of words); collocational features capture the syntactic environment of the target word and are usually represented by a small number of words and/or part-of-speech tags to the left or right of the target word.

The results obtained by a variety of classification methods are given in Table 6. All methods use either the full set or a subset of 18 confusion sets originally gathered by Golding (1995). Most methods are trained and tested on
i.e., words that are adjacent to the target word. For each word in the confusion set, we used the web to estimate how frequently it co-occurs with a word or a pair of words immediately to its left or right. Disambiguation is then performed by selecting the word in the confusion set with the highest co-occurrence frequency or probability. The baseline (f(t)) in Table 5 was obtained by always choosing the most frequent unigram in the confusion set. We used the same test set (2056 tokens) and the BNC corpus (Marcus et al., 1993) for testing.

Table 5: Performance of Altavista counts and BNC counts for context sensitive spelling correction (data from Cucerzan and Yarowsky 2002)

<table>
<thead>
<tr>
<th>Model</th>
<th>Alta</th>
<th>BNC</th>
<th>Model</th>
<th>Alta</th>
<th>BNC</th>
</tr>
</thead>
<tbody>
<tr>
<td>f(t)</td>
<td>72.98</td>
<td>70.00</td>
<td>f(w1,t,w2)/f(t)</td>
<td>87.77</td>
<td>76.53</td>
</tr>
<tr>
<td>f(w1,t)</td>
<td>84.40</td>
<td>83.02</td>
<td>f(w1,t,w2)/f(t)</td>
<td>86.27</td>
<td>74.47</td>
</tr>
<tr>
<td>f(t,w1)</td>
<td>84.89</td>
<td>82.74</td>
<td>f(t,w1,w2)/f(t)</td>
<td>84.94</td>
<td>74.23</td>
</tr>
<tr>
<td>f(w1,t,w2)</td>
<td>89.24#</td>
<td>77.13</td>
<td>f(w1,t,w2)/f(w1,t)</td>
<td>80.70</td>
<td>73.69</td>
</tr>
<tr>
<td>f(w1,w2,t)</td>
<td>87.13</td>
<td>74.89</td>
<td>f(w1,t,w2)/f(t,w2)</td>
<td>82.24</td>
<td>75.10</td>
</tr>
<tr>
<td>f(t,w1,w2)</td>
<td>84.68</td>
<td>75.08</td>
<td>f(w1,w2,t)/f(t,w2)</td>
<td>72.11</td>
<td>69.28</td>
</tr>
<tr>
<td>f(w1,t)/f(t)</td>
<td>82.81</td>
<td>77.84</td>
<td>f(w1,t,w1)/f(t,w1)</td>
<td>75.65</td>
<td>72.57</td>
</tr>
<tr>
<td>f(t,w1)/f(t)</td>
<td>77.49</td>
<td>80.71#</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Performance comparison with the literature for context sensitive spelling correction

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline BNC</td>
<td>70.00</td>
</tr>
<tr>
<td>Baseline Altavista</td>
<td>72.98</td>
</tr>
<tr>
<td>Best BNC</td>
<td>80.71†‡</td>
</tr>
<tr>
<td>Golding (1995)</td>
<td>81.40</td>
</tr>
<tr>
<td>Jones and Martin (1997)</td>
<td>84.26</td>
</tr>
<tr>
<td>Best Altavista</td>
<td>89.24†‡</td>
</tr>
<tr>
<td>Golding and Schabes (1996)</td>
<td>89.82</td>
</tr>
<tr>
<td>Mangu and Brill (1997)</td>
<td>92.79</td>
</tr>
<tr>
<td>Cucerzan and Yarowsky (2002)</td>
<td>92.20</td>
</tr>
<tr>
<td>Golding and Roth (1999)</td>
<td>94.23</td>
</tr>
</tbody>
</table>

5 Ordering of Prenominal Adjectives

The ordering of prenominal modifiers is important for natural language generation systems where the text must be both fluent and grammatical. For example, the sequence big fat Greek wedding is perfectly acceptable, whereas fat Greek big wedding sounds odd. The ordering of prenominal adjectives has sparked a great deal of theoretical debate (see Shaw and Hatzivassiloglou 1999 for an overview) and efforts have concentrated on defining rules based on semantic criteria that account for different orders (e.g., age < color, value < dimension).

Data intensive approaches to the ordering problem rely on corpora for gathering evidence for the likelihood of different orders. They rest on the hypothesis that the relative order of premodifiers is fixed, and independent of context and the noun being modified. The simplest strategy is what Shaw and Hatzivassiloglou (1999) call direct evidence. Given an adjective pair (a, b), they count how many times (a, b) and (b, a) appear in the corpus and choose the pair with the highest frequency.

Unfortunately the direct evidence method performs poorly when a given order is unseen in the training data. To compensate for this, Shaw and Hatzivassiloglou (1999) propose to compute the transitive closure of the ordering relation: if a < c and c < b, then a < b. Malouf (2000) further proposes a back-off bigram model of adjective pairs for choosing among alternative orders (P(⟨a,b⟩|{a,b}) vs. P(⟨b,a⟩|{a,b})). He also proposes positional probabilities as a means of estimating how likely it is for a given adjective a to appear first in a sequence by looking at each pair in the training data that contains the adjective a and recording its position. Finally, he uses memory-based learning as a means to encode morphological and semantic similarities among different adjective orders. Each adjective pair ab is encoded as a vector of 16 features (the last eight characters of a and the last eight characters of b) and a class ⟨a,b⟩ or

3An exception is Golding (1995), who uses the entire Brown corpus for training (1M words) and 344 of the Wall Street Journal corpus (Marcus et al., 1993) for testing.
Malouf (2000) extracted 263,838 individual pairs of adjectives from the BNC which he randomly partitioned into test (10%) and training data (90%) and evaluated all the above methods for ordering prenominal adjectives. His results showed that a memory-based classifier that uses morphological information as well as positional probabilities as features outperforms all other methods (see Table 7). For the ordering task we restricted ourselves to the direct evidence strategy which simply chooses the adjective order with the highest frequency or probability (see Table 7). Web counts were obtained by submitting literal queries to Altavista (see Section 2). We used the same 263,838 adjective pairs that Malouf extracted from the BNC. These were randomly partitioned into a training (90%) and test corpus (10%). The test corpus contained 26,271 adjective pairs. Given that submitting 26,271 queries to Altavista would be fairly time-consuming, a random sample of 1000 sequences was obtained from the test corpus and the web frequencies of these pairs were retrieved. The best Altavista model significantly outperformed the best BNC model, as indicated in Table 7. We also found that there was no significant difference between the best Altavista model and the best model reported by Malouf, a supervised method using positional probability estimates from the BNC and morphological variants.

## 6 Bracketing of Compound Nouns

The first analysis task we consider is the syntactic disambiguation of compound nouns, which has received a fair amount of attention in the NLP literature (Pustejovsky et al., 1993; Resnik, 1993; Lauer, 1995). The task can be summarized as follows: given a three word compound \( n_1 n_2 n_3 \), determine the correct binary bracketing of the word sequence (see (3) for an example).

\[
\langle h, a \rangle.
\]

(a) [[backup compiler] disk]

(b) backup [compiler disk]

Previous approaches typically compare different bracketings and choose the most likely one. The \textbf{adjacency model} compares \( [n_1 n_2] \) against \( [n_2 n_3] \) and adopts a right branching analysis if \( [n_2 n_3] \) is more likely than \( [n_1 n_2] \). The \textbf{dependency model} compares \( [n_1 n_2] \) against \( [n_1 n_3] \) and adopts a right branching analysis if \( [n_1 n_3] \) is more likely than \( [n_1 n_2] \).

The simplest model of compound noun disambiguation compares the frequencies of the two competing analyses and opts for the most frequent one (Pustejovsky et al.,

<table>
<thead>
<tr>
<th>Model</th>
<th>Altavista</th>
<th>BNC</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f(a_1, a_2) : f(a_2, a_1) )</td>
<td>89.6%</td>
<td>80.4%</td>
</tr>
<tr>
<td>( f(a_1, a_2) / f(a_2) : f(a_2, a_1) / f(a_1) )</td>
<td>83.2%</td>
<td>77.0%</td>
</tr>
<tr>
<td>( f(a_1, a_2) / f(a_1) : f(a_2, a_1) / f(a_2) )</td>
<td>80.2%</td>
<td>80.6%</td>
</tr>
</tbody>
</table>

Malouf (2000); memory-based - 91.0%

Table 7: Performance of Altavista counts and BNC counts for adjective ordering (data from Malouf 2000)

<table>
<thead>
<tr>
<th>Model</th>
<th>Altavista</th>
<th>BNC</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f(n_1, n_2) : f(n_2, n_3) )</td>
<td>65.57</td>
<td>63.93</td>
</tr>
<tr>
<td>( f(n_1, n_2) : f(n_1, n_3) )</td>
<td>68.85</td>
<td>65.57</td>
</tr>
<tr>
<td>( f(n_1, n_2) / f(n_1) : f(n_2, n_3) / f(n_3) )</td>
<td>70.49</td>
<td>63.11</td>
</tr>
<tr>
<td>( f(n_1, n_2) / f(n_2) : f(n_1, n_3) / f(n_3) )</td>
<td>80.32</td>
<td>66.39</td>
</tr>
<tr>
<td>( f(n_1, n_2) : f(n_1, n_3) (NEAR) )</td>
<td>68.03</td>
<td>63.11</td>
</tr>
<tr>
<td>( f(n_1, n_2) : f(n_1, n_3) (NEAR) )</td>
<td>71.31</td>
<td>67.21</td>
</tr>
<tr>
<td>( f(n_1, n_2) / f(n_1) : f(n_2, n_3) / f(n_3) (NEAR) )</td>
<td>61.47</td>
<td>62.29</td>
</tr>
<tr>
<td>( f(n_1, n_2) : f(n_1, n_3) (NEAR) )</td>
<td>65.57</td>
<td>57.37</td>
</tr>
<tr>
<td>( f(n_1, n_2) : f(n_1, n_3) / f(n_2, n_3) / f(n_3) (NEAR) )</td>
<td>75.40</td>
<td>68.03%</td>
</tr>
</tbody>
</table>

Lauer (1995): dependency 77.50

Table 8: Performance of Altavista counts and BNC counts for compound bracketing (data from Lauer 1995)

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>63.93</td>
</tr>
<tr>
<td>Best BNC</td>
<td>68.03†‡</td>
</tr>
<tr>
<td>Lauer (1995): adjacency</td>
<td>68.90</td>
</tr>
<tr>
<td>Lauer (1995): dependency</td>
<td>77.50</td>
</tr>
<tr>
<td>Best Altavista</td>
<td>78.68†‡</td>
</tr>
<tr>
<td>Lauer (1995): tuned</td>
<td>80.70</td>
</tr>
<tr>
<td>Upper bound</td>
<td>81.50</td>
</tr>
</tbody>
</table>

Table 9: Performance comparison with the literature for compound bracketing

1993). Lauer (1995) proposes an unsupervised method for estimating the frequencies of the competing bracketings based on a taxonomy or a thesaurus. He uses a probability ratio to compare the probability of the left-branching analysis to that of the right-branching (see (4) for the dependency model and (5) for the adjacency model).

\[
R_{dep} = \frac{\sum_{t_i \in cat(w_i)} P(t_1 \rightarrow t_2) P(t_2 \rightarrow t_3)}{\sum_{t_i \in cat(w_i)} P(t_1 \rightarrow t_3) P(t_2 \rightarrow t_3)}
\]

(4)

\[
R_{adj} = \frac{\sum_{t_i \in cat(w_i)} P(t_1 \rightarrow t_2)}{\sum_{t_i \in cat(w_i)} P(t_2 \rightarrow t_3)}
\]

(5)

Here \( t_1, t_2 \) and \( t_3 \) are conceptual categories in the taxonomy or thesaurus, and the nouns \( w_1 \ldots w_i \) are members of these categories. The estimation of probabilities over concepts (rather than words) reduces the number of model parameters and effectively decreases the amount of training data required. The probability \( P(t_1 \rightarrow t_2) \) denotes the modification of a category \( a \) by a category \( t_1 \).

Lauer (1995) tested both the adjacency and dependency models on 244 compounds extracted from Grolier’s encyclopedia, a corpus of 8 million words. Frequencies for the two models were obtained from the same corpus and from Roget’s thesaurus (version 1911) by counting pairs of nouns that are either strictly adjacent or co-occur within a window of a fixed size (e.g., two, three, fi, fy, or hundred words). The majority of the bracketings in our test set were left-branching, yielding a baseline of 63.93% (see Table 9). Lauer’s best results (77.50%) were obtained with the dependency model and a training
scheme which takes strictly adjacent nouns into account. Performance increased further by 3.2% when POS tags were taken into account. The results for this tuned model are also given in Table 9. Finally, Lauer conducted an experiment with human judges to assess the upper bound for the bracketing task. An average accuracy of 81.50% was obtained.

We replicated Lauer’s (1995) results for compound noun bracketing using the same test set. We compared the performance of the adjacency and dependency models (see (4) and (5)), but instead of relying on a corpus and a thesaurus, we estimated the relevant probabilities using web counts. The latter were obtained using inflected queries (see Section 2) and Altavista’s NEAR operator. Ties were resolved by defaulting to the most frequent form of the word (rather than concept) frequencies which Lauer estimated from the web using inflected queries (see Section 2). Co-occurrence frequencies were estimated from the web-based trigram model (see (6)) where probabilities are calculated on the basis of word (rather than concept) frequencies which Lauer obtained from Grolier’s encyclopedia heuristically via pattern matching.

Lauer (1995) also experimented with a lexicalized version of (7) where probabilities are calculated on the basis of word (rather than concept) frequencies which Lauer obtained from Grolier’s encyclopedia heuristically via pattern matching.

Lauer (1995) tested the model in (7) on 282 compounds that he selected randomly from Grolier’s encyclopedia and annotated with their paraphrasing prepositions. The preposition of accounted for 33% of the paraphrases in this data set (see Baseline in Table 11). The concept-based model (see (7)) achieved an accuracy of 48% on this test set, whereas its lexicalized version reached an accuracy of 40% (see Table 11).

We attempted the interpretation task with the lexicalized version of the bigram model (see (7)), but also tried the more data intensive trigram model (see (6)), again in its lexicalized form. Furthermore, we experimented with several conditional and unconditional variants of (7) and (6). Co-occurrence frequencies were estimated from the web using inflected queries (see Section 2). Determiners were inserted before nouns resulting in queries of the type story/stories about and about the/a/0 war/wars for the compound war story. As shown in Table 10, the best performance was obtained using the web-based trigram model (f(n_1, p, n_2)); it signifi cantly outperformed the best BNC model. The comparison with the literature in Table 11 showed that the best Altavista model signifi cantly outperformed the best Altavista model (see (7) and (6)). Co-occurrence frequencies were estimated from the web using inflected queries (see Section 2). Determiners were inserted before nouns resulting in queries of the type story/stories about and about the/a/0 war/wars for the compound war story.

### Table 10: Performance of Altavista counts and BNC counts for compound interpretation (data from Lauer 1995)

<table>
<thead>
<tr>
<th>Model</th>
<th>Altavista</th>
<th>BNC</th>
</tr>
</thead>
<tbody>
<tr>
<td>f(n_1, p)f(p, n_2)</td>
<td>30.71</td>
<td>27.85*</td>
</tr>
<tr>
<td>f(n_1, p, n_2)</td>
<td>55.71#†‡</td>
<td>11.42</td>
</tr>
<tr>
<td>f(n_1, p)f(p, n_2)/f(p)</td>
<td>47.14</td>
<td>26.42</td>
</tr>
<tr>
<td>f(n_1, p, n_2)/f(p)</td>
<td>55.00</td>
<td>10.71</td>
</tr>
</tbody>
</table>

### Table 11: Performance comparison with the literature for compound interpretation

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lauer (1995): concept-based</td>
<td>28.00</td>
</tr>
<tr>
<td>Baseline</td>
<td>33.00</td>
</tr>
<tr>
<td>Lauer (1995): word-based</td>
<td>40.00</td>
</tr>
<tr>
<td>Best Altavista</td>
<td>55.71†‡</td>
</tr>
</tbody>
</table>

Here, t_1 and t_2 represent concepts in Roget’s thesaurus. Lauer (1995) also experimented with a lexicalized version of (7) where probabilities are calculated on the basis of word (rather than concept) frequencies which Lauer obtained from Grolier’s encyclopedia heuristically via pattern matching.

Lauer (1995) is the first to propose and evaluate an unsupervised probabilistic model of compound noun interpretation for domain independent text. By recasting the interpretation problem in terms of paraphrasing, Lauer assumes that the semantic relations of compound heads and modifiers can be expressed via prepositions that (in contrast to abstract semantic relations) can be found in a corpus. For example, in order to interpret war story, one needs to find in a corpus related paraphrases: story about the war, story of the war, war story, etc. Lauer uses eight prepositions for the paraphrasing task (of, for, in, at, on, from, with, about). A simple model of compound noun paraphrasing is shown in (6):

\[
p^* = \arg \max_p P(p|n_1, n_2)
\]

Lauer (1995) points out that the above model contains one parameter for every triple \( (p, n_1, n_2) \), and as a result hundreds of millions of training instances would be necessary. As an alternative to (6), he proposes the model in (7) which combines the probability of the modifier given a certain preposition with the probability of the head given the same preposition, and assumes that these two probabilities are independent.

\[
p^* = \arg \max_p \sum_{t_1, t_2 \in \text{cats}(n_1)} \sum_{t_2 \in \text{cats}(n_2)} P(t_1|p)P(t_2|p)
\]

Here, t_1 and t_2 represent concepts in Roget’s thesaurus. Lauer (1995) also experimented with a lexicalized version of (7) where probabilities are calculated on the basis of word (rather than concept) frequencies which Lauer obtained from Grolier’s encyclopedia heuristically via pattern matching.

### Interpretation of Compound Nouns

The second analysis task we consider is the semantic interpretation of compound nouns. Most previous approaches to this problem have focused on the interpretation of two word compounds whose nouns are related via a basic set of semantic relations (e.g., CAUSE relates onion tears, FOR relates pet spray). The majority of proposals are symbolic and therefore limited to a specific domain due to the large effort involved in hand-coding semantic information (see Lauer 1995 for an extensive overview).

Lauer (1995) tested the model in (7) on 282 compounds that he selected randomly from Grolier’s encyclopedia and annotated with their paraphrasing prepositions. The preposition of accounted for 33% of the paraphrases in this data set (see Baseline in Table 11). The concept-based model (see (7)) achieved an accuracy of 48% on this test set, whereas its lexicalized version reached an accuracy of 40% (see Table 11).
achieved a performance that is not signifi cantly different from the baseline, and signifi cantly worse than Lauer’s best model.

8 Noun Countability Detection

The next analysis task that we consider is the problem of determining the countability of nouns. Countability is the semantic property that determines whether a noun can occur in singular and plural forms, and affects the range of permissible modifi ers. In English, nouns are typically either countable (e.g., *one dog, two dogs*) or uncountable (e.g., *some peace, *one peace, *two peaces*).

Baldwin and Bond (2003) propose a method for automatically learning the countability of English nouns from the BNC. They obtain information about noun countability by merging lexical entries from COMLEX (Grishman et al., 1994) and the ALTJ/E Japanese-to-English semantic transfer dictionary (Ikehara et al., 1991). Words are classifi ed into four classes: countable, uncountable, bi-partite (e.g., *tousers*), and plural only (e.g., *goods*). A memory-based classifi er is used to learn the four-way distinction on the basis of several linguistically motivated features such as: number of the head noun, number of the modifi er, subject-verb agreement, plural determiners.

We devised unsupervised models for the countability learning task and evaluated their performance on Baldwin and Bond’s (2003) test data. We concentrated solely on countable and uncountable nouns, as they account for the vast majority of the data. Four models were tested: (a) compare the frequency of the singular and plural forms of the noun; (b) compare the frequency of determiner-noun pairs that are characteristic of countable or uncountable nouns; the determiners used were *many* for countable and *much* for uncountable ones; (c) same as model (b), but the det-noun frequencies are normalized by the frequency of the noun; (d) backoff: try to make a decision using det-noun frequencies; if these are too sparse, back off to singular/plural frequencies.

Unigram and bigram frequencies were estimated from the web using literal queries; for models (a)–(c) a threshold parameter was optimized on the development set (this parameter determines the ratio of singular/plural frequencies or det-noun frequencies above which a noun was considered as countable). For model (b), an additional backoff parameter was used, specifying the minimum frequency that triggers backoff.

The models and their performance on the test set are listed in Table 12. The best Altavista model is the conditional det-noun model \( f(det,n)/f(n) \), which achieves 88.38% on countable and 91.22% on uncountable nouns. On the BNC, the simple unigram model performs best. Its performance is not statistically different from that of the best Altavista model. Note that for the BNC models, data sparseness means the det-noun models perform poorly, which is why the backoff model was not attempted here. Table 13 shows that both the Altavista and the BNC model perform signifi cantly worse than the best model proposed by Baldwin and Bond (2003); this is a supervised model that uses many more features than just singular/plural frequency and det-noun frequency.

9 Conclusions

We showed that simple, unsupervised models using web counts can be devised for a variety of NLP tasks. The tasks were selected so that they cover both syntax and semantics, both generation and analysis, and a wider range of \( n \)-grams than have been previously used.

For all but two tasks (candidate selection for MT and noun countability detection) we found that simple, unsupervised models perform signifi cantly better when \( n \)-gram frequencies are obtained from the web rather than from a standard large corpus. This result is consistent with Keller and Lapata’s (2003) fi ndings that the web yields better counts than the BNC. The reason for this seems to be that the web is much larger than the BNC (about 1000 times); the size seems to compensate for the fact that simple heuristics were used to obtain web counts, and for the noise inherent in web data.

Our results were less encouraging when it comes to comparisons with state-of-the-art models. We found that in all but one case, web-based models fail to signifi cantly outperform the state of the art. The exception was compound noun interpretation, for which the Altavista model was signifi cantly better than the Lauer’s (1995) model. For three tasks (candidate selection for MT, adjective ordering, and compound noun bracketing), we found that the performance of the web-based models was not signifi cantly different from the performance of the best models reported in the literature.

Note that for all the tasks we investigated, the best performance in the literature was obtained by supervised models that have access not only to simple bigram or tri-
gram frequencies, but also to linguistic information such as part-of-speech tags, semantic restrictions, or context (or a thesaurus, in the case of Lauer’s models). When unsupervised web-based models are compared against supervised methods that employ a wide variety of features, we observe that having access to linguistic information makes up for the lack of vast amounts of data.

Our results therefore indicate that large data sets such as those obtained from the web are not the panacea that they are claimed to be (at least implicitly) by authors such as Grefenstette (1998) and Keller and Lapata (2003). Rather, in our opinion, web-based models should be used as a new baseline for NLP tasks. The web baseline indicates how much can be achieved with a simple, unsupervised model based on n-grams with access to a huge data set. This baseline is more realistic than baselines obtained from standard corpora; it is generally harder to beat, as our comparisons with the BNC baseline throughout this paper have shown.

Note that for certain tasks, the performance of a web baseline model might actually be sufficient, so that the effort of constructing a sophisticated supervised model and annotating the necessary training data can be avoided. Another possibility that needs further investigation is the combination of web-based models with supervised methods. This can be done with ensemble learning methods or simply by using web-based frequencies (or probabilities) as features (in addition to linguistically motivated features) to train supervised classifiers.

Acknowledgments

We are grateful to Tim Baldwin, Silviu Cucerzan, Mark Lauer, Rob Malouf, Detelef Prescher, and Adwait Ratnaparkhi for making their data sets available.

References


