Search Engine Statistics Beyond the n-gram: Application to Noun Compound Bracketing

Preslav Nakov
EECS, Computer Science Division
University of California, Berkeley
Berkeley, CA 94720
nakov@cs.berkeley.edu

Marti Hearst
SIMS
University of California, Berkeley
Berkeley, CA 94720
hearst@sims.berkeley.edu

Abstract

In order to achieve the long-range goal of semantic interpretation of noun compounds, it is often necessary to first determine their syntactic structure. This paper describes an unsupervised method for noun compound bracketing which extracts statistics from Web search engines using a χ^2 measure, a new set of surface features, and paraphrases. On a gold standard, the system achieves results of 89.34% (baseline 66.80%), which is a sizable improvement over the state of the art (80.70%).

1 Introduction

An important but understudied language analysis problem is that of noun compound bracketing, which is generally viewed as a necessary step towards noun compound interpretation. 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*. In order to make these semantic distinctions accurately, it can be useful to begin with the correct grouping of terms, since choosing a particular syntactic structure limits the options left for semantics. 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)

In this paper, we describe a highly accurate unsupervised method for making bracketing decisions for noun compounds (NCs). We improve on the current standard approach of using bigram estimates to compute adjacency and dependency scores by introducing the use of the χ^2 measure for this problem. We also introduce a new set of surface features for querying Web search engines which prove highly effective. Finally, we experiment with paraphrases for improving prediction statistics. We have evaluated the application of combinations of these features to predict NC bracketing on two distinct collections, one consisting of terms drawn from encyclopedia text, and another drawn from bioscience text.

The remainder of this paper describes related work, the word association models, the surface features, the paraphrase features and the results.

2 Related Work

The syntax and semantics of NCs is an active area of research; the *Journal of Computer Speech and Language* has an upcoming special issue on Multiword Expressions.

The best known early work on automated unsupervised NC bracketing is that of Lauer (1995) who introduces the probabilistic dependency model for the syntactic disambiguation of NCs and argues against the adjacency model, proposed by Marcus (1980), Pustejovsky et al. (1993) and Resnik (1993). Lauer collects n-gram statistics from Grolier's encyclopedia, containing about 8 million words. To

Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL),
pages 17–24, Ann Arbor, June 2005.

overcome data sparsity problems, he estimates probabilities over conceptual categories in a taxonomy (Roget's thesaurus) rather than for individual words.

Lauer evaluated his models on a set of 244 unambiguous NCs derived from the same encyclopedia (inter-annotator agreement 81.50%) and achieved 77.50% for the dependency model above (baseline 66.80%). Adding POS and further tuning allowed him to achieve the state-of-the-art result of 80.70%.

More recently, Keller and Lapata (2003) evaluate the utility of using Web search engines for obtaining frequencies for unseen bigrams. They then later propose using Web counts as a baseline unsupervised method for many NLP tasks (Lapata and Keller, 2004). They apply this idea to six NLP tasks, including the syntactic and semantic disambiguation of NCs following Lauer (1995), and show that variations on bigram counts perform nearly as well as more elaborate methods. They do not use taxonomies and work with the word n-grams directly, achieving 78.68% with a much simpler version of the dependency model.

Girju et al. (2005) propose a *supervised* model (decision tree) for NC bracketing *in context*, based on five semantic features (requiring the correct WordNet sense to be given): the top three WordNet semantic classes for each noun, derivationally related forms and whether the noun is a nominalization. The algorithm achieves accuracy of 83.10%.

3 Models and Features

3.1 Adjacency and Dependency Models

In related work, a distinction is often made between what is called the *dependency model* and the *adjacency model*. 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).

Left bracketing is a bit different since there is only modificational choice for a 3-word NC. If w_1 modifies w_2, this implies that w_1w_2 is a compound which in turn modifies w_3, as in *law enforcement agent*.

Thus the usefulness of the adjacency model vs. the dependency model can depend in part on the mix of left and right bracketing. Below we show that the dependency model works better than the adjaceny model, confirming other results in the literature. The next subsections describe several different ways to compute these measures.

3.2 Using Frequencies

The most straightforward way to compute adjacency and dependency scores is to simply count the corresponding frequencies. Lapata and Keller (2004) achieved their best accuracy (78.68%) with the dependency model and the simple symmetric score $\#(w_i, w_j)$.[1]

3.3 Computing Probabilities

Lauer (1995) assumes that adjacency and dependency should be computed via probabilities. Since they are relatively simple to compute, we investigate them in our experiments.

Consider the dependency model, as introduced above, and the NC $w_1w_2w_3$. Let $\Pr(w_i \rightarrow w_j|w_j)$ be the probability that the word w_i precedes a given fixed word w_j. Assuming that the distinct head-modifier relations are independent, we obtain $\Pr(\text{right}) = \Pr(w_1 \rightarrow w_3|w_3)\Pr(w_2 \rightarrow w_3|w_3)$ and $\Pr(\text{left}) = \Pr(w_1 \rightarrow w_2|w_2)\Pr(w_2 \rightarrow w_3|w_3)$. To choose the more likely structure, we can drop the shared factor and 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

[1] This score worked best on training, when Keller&Lapata were doing model selection. On testing, $\Pr$ (with the dependency model) worked better and achieved accuracy of 80.32%, but this result was ignored, as $\Pr$ did worse on training.

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[2] each of these frequencies by adding 0.5 to avoid problems caused by nonexistent n-grams.

Unless some particular probabilistic interpretation is needed,[3] there is no reason why for a given ordered pair of words (w_i, w_j), we should use $\Pr(w_i \rightarrow w_j|w_j)$ rather than $\Pr(w_j \rightarrow w_i|w_i)$, $i < j$. This is confirmed by the adjacency model experiments in (Lapata and Keller, 2004) on Lauer's NC set. Their results show that both ways of computing the probabilities make sense: using Altavista queries, the former achieves a higher accuracy (70.49% vs. 68.85%), but the latter is better on the British National Corpus (65.57% vs. 63.11%).

3.4 Other Measures of Association

In both models, the probability $\Pr(w_i \rightarrow w_j|w_j)$ can be replaced by some (possibly symmetric) measure of association between w_i and w_j, such as *Chi squared* (χ^2). To calculate $\chi^2(w_i, w_j)$, we need:

- (A) $\#(w_i, w_j)$;
- (B) $\#(w_i, \overline{w_j})$, the number of bigrams in which the first word is w_i, followed by a word other than w_j;
- (C) $\#(\overline{w_i}, w_j)$, the number of bigrams, ending in w_j, whose first word is other than w_i;
- (D) $\#(\overline{w_i}, \overline{w_j})$, the number of bigrams in which the first word is not w_i and the second is not w_j.

They are combined in the following formula:

$$\chi^2 = \frac{N(AD - BC)^2}{(A + C)(B + D)(A + B)(C + D)} \quad (1)$$

Here $N = A + B + C + D$ is the total number of bigrams, $B = \#(w_i) - \#(w_i, w_j)$ and $C = \#(w_j) - \#(w_i, w_j)$. While it is hard to estimate D directly, we can calculate it as $D = N - A - B - C$. Finally, we estimate N as the total number of indexed bigrams on the Web. They are estimated as 8 trillion, since Google indexes about 8 billion pages and each contains about 1,000 words on average.

Other measures of word association are possible, such as *mutual information* (MI), which we can use with the dependency and the adjacency models, similarly to #, χ^2 or Pr. However, in our experiments, χ^2 worked better than other methods; this is not surprising, as χ^2 is known to outperform MI as a measure of association (Yang and Pedersen, 1997).

3.5 Web-Derived Surface Features

Authors sometimes (consciously or not) disambiguate the words they write by using surface-level markers to suggest the correct meaning. We have found that exploiting these markers, when they occur, can prove to be very helpful for making bracketing predictions. The enormous size of Web search engine indexes facilitates finding such markers frequently enough to make them useful.

One very productive feature is the *dash* (hyphen). Starting with the term *cell cycle analysis*, if we can find a version of it in which a dash occurs between the first two words: *cell-cycle*, this suggests a left bracketing for the full NC. Similarly, the dash in *donor T-cell* favors a right bracketing. The right-hand dashes are less reliable though, as their scope is ambiguous. In *fiber optics-system*, the hyphen indicates that the noun compound *fiber optics* modifies *system*. There are also cases with multiple hyphens, as in *t-cell-depletion*, which preclude their use.

The genitive ending, or *possessive* marker is another 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.[4]

Another highly reliable source is related to internal *capitalization*. For example *Plasmodium vivax Malaria* suggests left bracketing, while *brain Stem cells* would favor a right one. (We disable this feature on Roman digits and single-letter words to prevent problems with terms like *vitamin D deficiency*, where the capitalization is just a convention as opposed to a special mark to make the reader think that the last two terms should go together.)

[2] 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.

[3] For example, as used by Lauer to introduce a prior for left-right bracketing preference. The best Lauer model does not work with words directly, but uses a taxonomy and further needs a probabilistic interpretation so that the hidden taxonomy variables can be summed out. Because of that summation, the term $\Pr(w_2 \rightarrow w_3|w_3)$ does not cancel in his dependency model.

[4] Features can also occur combined, e.g. *brain's stem-cells*.

We can also make use of embedded *slashes*. For example in *leukemia/lymphoma cell*, the slash predicts a right bracketing since the first word is an alternative and cannot be a modifier of the second one.

In some cases we can find instances of the NC in which one or more words are enclosed in parentheses, e.g., *growth factor (beta)* or *(growth factor) beta*, both of which indicate a left structure, or *(brain) stem cells*, which suggests a right bracketing.

Even a comma, a dot or a colon (or any special character) can act as indicators. For example, "*health care, provider*" or "*lung cancer: patients*" are weak predictors of a left bracketing, showing that the author chose to keep two of the words together, separating out the third one.

We can also exploit dashes to words external to the target NC, as in $mouse$*-brain stem cells*, which is a weak indicator of right bracketing.

Unfortunately, Web search engines ignore punctuation characters, thus preventing querying directly for terms containing hyphens, brackets, apostrophes, etc. We collect them indirectly by issuing queries with the NC as an exact phrase and then post-processing the resulting summaries, looking for the surface features of interest. Search engines typically allow the user to explore up to 1000 results. We collect all results and summary texts that are available for the target NC and then search for the surface patterns using regular expressions over the text. Each match increases the score for left or right bracketing, depending on which the pattern favors.

While some of the above features are clearly more reliable than others, we do not try to weight them. For a given NC, we post-process the returned Web summaries, then we find the number of left-predicting surface feature instances (regardless of their type) and compare it to the number of right-predicting ones to make a bracketing decision.[5]

3.6 Other Web-Derived Features

Some features can be obtained by using the overall counts returned by the search engine. As these counts are derived from the entire Web, as opposed to a set of up to 1,000 summaries, they are of different magnitude, and we did not want to simply add them to the surface features above. They appear as independent models in Tables 1 and 2.

First, in some cases, we can query for *possessive markers* 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 vs. 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 (TN) factor"* would favor left. We would like to issue exact phrase queries for the two 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 Roman digit (e.g., *IV*), a state (e.g., *CA*), 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"*).

We also used Google's support for "*", which allows a single word wildcard, to see how often two of the words are present but separated from the third by some other word(s). This implicitly tries to capture paraphrases involving the two sub-concepts making up the whole. For example, we compared the frequency of *"health care * reform"* to that of *"health * care reform"*. We also used 2 and 3 stars and switched the word group order (indicated with *rev.* in Tables 1 and 2), e.g., *"care reform * * health"*. We also tried a simple *reorder* without inserting stars, i.e., compare the frequency of *"reform health*

[5] This appears as *Surface features (sum)* in Tables 1 and 2.

care" to that of "*care reform health*". For example, when analyzing *myosin heavy chain* we see that *heavy chain myosin* is very frequent, which provides evidence against grouping *heavy* and *chain* together as they can commute.

Further, we tried 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 kinases activation"*. Alternatively, if we find different internal inflections of the first word, this would favor a right bracketing.

Finally, we tried 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"*.

3.7 Paraphrases

Warren (1978) proposes that the semantics of the relations between words in a noun compound are often made overt by paraphrase. As an example of *prepositional paraphrase*, an author describing the concept of *brain stem cells* may choose to write it in a more expanded manner, such as *stem cells in the brain*. This contrast can be helpful for syntactic bracketing, suggesting that the full NC takes on right bracketing, since *stem* and *cells* are kept together in the expanded version. However, this NC is ambiguous, and can also be paraphrased as *cells from the brain stem*, implying a left bracketing.

Some NCs' meaning cannot be readily expressed with a prepositional paraphrase (Warren, 1978). An alternative is the *copula paraphrase*, as in *office building that/which is a skyscraper* (right bracketing), or a *verbal paraphrase* such as *pain associated with arthritis migraine* (left).

Other researchers have used prepositional paraphrases as a proxy for determining the semantic relations that hold between nouns in a compound (Lauer, 1995; Keller and Lapata, 2003; Girju et al., 2005). Since most NCs have a prepositional paraphrase, Lauer builds a model trying to choose between the most likely candidate prepositions: *of*, *for*, *in*, *at*, *on*, *from*, *with* and *about* (excluding *like* which is mentioned by Warren). This could be problematic though, since as a study by Downing (1977) shows, when no context is provided, people often come up with incompatible interpretations.

In contrast, we use paraphrases in order to make syntactic bracketing assignments. Instead of trying to manually decide the correct paraphrases, we can issue queries using paraphrase patterns and find out how often each occurs in the corpus. We then add up the number of hits predicting a left versus a right bracketing and compare the counts.

Unfortunately, search engines lack linguistic annotations, making general verbal paraphrases too expensive. Instead we used a small set of hand-chosen paraphrases: *associated with*, *caused by*, *contained in*, *derived from*, *focusing on*, *found in*, *involved in*, *located at/in*, *made of*, *performed by*, *preventing*, *related to* and *used by/in/for*. It is however feasible to generate queries predicting left/right bracketing with/without a determiner for every preposition.[6] For the copula paraphrases we combine two verb forms *is* and *was*, and three complementizers *that*, *which* and *who*. These are optionally combined with a preposition or a verb form, e.g. *themes that are used in science fiction*.

4 Evaluation

4.1 Lauer's Dataset

We experimented with the dataset from (Lauer, 1995), in order to produce results comparable to those of Lauer and Keller & Lapata. The set consists of 244 unambiguous 3-noun NCs extracted from *Grolier's encyclopedia*; however, only 216 of these NCs are unique.

Lauer (1995) derived n-gram frequencies from the *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). Lapata and Keller (2004) derived their statistics from the Web and achieved results close to Lauer's using simple lexical models.

4.2 Biomedical Dataset

We constructed a new set of noun compounds from the biomedical literature. Using the Open NLP

[6] In addition to the articles (*a*, *an*, *the*), we also used quantifiers (e.g. *some*, *every*) and pronouns (e.g. *this*, *his*).

tools,[7] we sentence splitted, tokenized, POS tagged and shallow parsed a set of 1.4 million MEDLINE abstracts (citations between 1994 and 2003). Then we extracted all 3-noun sequences falling in the last three positions of noun phrases (NPs) found in the shallow parse. If the NP contained other nouns, the sequence was discarded. This allows for NCs which are modified by adjectives, determiners, and so on, but prevents extracting 3-noun NCs that are part of longer NCs. For details, see (Nakov et al., 2005).

This procedure resulted in 418,678 different NC types. We manually investigated the most frequent ones, removing those that had errors in tokenization (e.g., containing words like *transplan* or *tation*), POS tagging (e.g., *acute lung injury*, where *acute* was wrongly tagged as a noun) or shallow parsing (e.g., *situ hybridization*, that misses *in*). We had to consider the first 843 examples in order to obtain 500 good ones, which suggests an extraction accuracy of 59%. This number is low mainly because the tokenizer handles dash-connected words as a single token (e.g. *factor-alpha*) and many tokens contained other special characters (e.g. *cd4+*), which cannot be used in a query against a search engine and had to be discarded.

The 500 NCs were annotated independently by two judges, one of which has a biomedical background; the other one was one of the authors. The problematic cases were reconsidered by the two judges and after agreement was reached, the set contained: 361 left bracketed, 69 right bracketed and 70 ambiguous NCs. The latter group was excluded from the experiments.[8]

We calculated the inter-annotator agreement on the 430 cases that were marked as unambiguous after agreement. Using the original annotator's choices, we obtained an agreement of 88% or 82%, depending on whether we consider the annotations, that were initially marked as ambiguous by one of the judges to be correct. The corresponding values for the kappa statistics were .606 (substantial agreement) and .442 (moderate agreement).

[7] http://opennlp.sourceforge.net/

[8] Two NCs can appear more than once but with a different inflection or with a different word variant, e.g,. *colon cancer cells* and *colon carcinoma cells*.

4.3 Experiments

The n-grams, surface features, and paraphrase counts were collected by issuing exact phrase queries, limiting the pages to English and requesting filtering of similar results.[9] For each NC, we generated all possible word inflections (e.g., *tumor* and *tumors*) and alternative word variants (e.g., *tumor* and *tumour*). For the biomedical dataset they were automatically obtained from the UMLS Specialist lexicon.[10] For Lauer's set we used Carroll's morphological tools.[11] For bigrams, we inflect only the second word. Similarly, for a prepositional paraphrase we generate all possible inflected forms for the two parts, before and after the preposition.

4.4 Results and Discussion

The results are shown in Tables 1 and 2. As NCs are left-bracketed at least 2/3rds of the time (Lauer, 1995), a straightforward baseline is to always assign a left bracketing. Tables 1 and 2 suggest that the surface features perform best. The paraphrases are equally good on the biomedical dataset, but on Lauer's set their performance is lower and is comparable to that of the dependency model.

The dependency model clearly outperforms the adjacency one (as other researchers have found) on Lauer's set, but not on the biomedical set, where it is equally good. χ^2 barely outperforms #, but on the biomedical set χ^2 is a clear winner (by about 1.5%) on both dependency and adjacency models.

The frequencies (#) outperform or at least rival the probabilities on both sets and for both models. This is not surprising, given the previous results by Lapata and Keller (2004). Frequencies also outperform Pr on the biomedical set. This may be due to the abundance of single-letter words in that set (because of terms like *T cell*, *B cell*, *vitamin D* etc.; similar problems are caused by Roman digits like *ii*, *iii* etc.), whose Web frequencies are rather unreliable, as they are used by Pr but not by frequencies. Single-letter words cause potential problems for the paraphrases

[9] In our experiments we used MSN Search statistics for the n-grams and the paraphrases (unless the pattern contained a "*"), and Google for the surface features. MSN always returned exact numbers, while Google and Yahoo rounded their page hits, which generally leads to lower accuracy (Yahoo was better than Google for these estimates).

[10] http://www.nlm.nih.gov/pubs/factsheets/umlslex.html

[11] http://www.cogs.susx.ac.uk/lab/nlp/carroll/morph.html

Model	√	×	∅	P(%)	C(%)
# adjacency	183	61	0	75.00	100.00
Pr adjacency	180	64	0	73.77	100.00
MI adjacency	182	62	0	74.59	100.00
χ^2 adjacency	184	60	0	**75.41**	100.00
# dependency	193	50	1	79.42	99.59
Pr dependency	194	50	0	79.51	100.00
MI dependency	194	50	0	79.51	100.00
χ^2 dependency	195	50	0	**79.92**	100.00
# adjacency (*)	152	41	51	78.76	79.10
# adjacency (**)	162	43	39	79.02	84.02
# adjacency (***)	150	51	43	74.63	82.38
# adjacency (*, rev.)	163	48	33	77.25	86.47
# adjacency (**, rev.)	165	51	28	76.39	88.52
# adjacency (***, rev.)	156	57	31	73.24	87.30
Concatenation adj.	175	48	21	78.48	91.39
Concatenation dep.	167	41	36	**80.29**	85.25
Concatenation triples	76	3	165	**96.20**	32.38
Inflection Variability	69	36	139	65.71	43.03
Swap first two words	66	38	140	63.46	42.62
Reorder	112	40	92	73.68	62.30
Abbreviations	21	3	220	**87.50**	9.84
Possessives	32	4	208	**88.89**	14.75
Paraphrases	174	38	32	**82.08**	86.89
Surface features (sum)	183	31	30	**85.51**	87.70
Majority vote	210	22	12	90.52	95.08
Majority vote → left	*218*	*26*	*0*	***89.34***	***100.00***
Baseline (choose left)	163	81	0	66.80	100.00

Table 1: **Lauer Set.** Shown are the numbers for correct (√), incorrect (×), and no prediction (∅), followed by precision (P, calculated over √ and × only) and coverage (C, % examples with prediction). We use "→" for back-off to another model in case of ∅.

Model	√	×	∅	P(%)	C(%)
# adjacency	374	56	0	86.98	100.00
Pr adjacency	353	77	0	82.09	100.00
MI adjacency	372	58	0	86.51	100.00
χ^2 adjacency	379	51	0	**88.14**	100.00
# dependency	374	56	0	86.98	100.00
Pr dependency	369	61	0	85.81	100.00
MI dependency	369	61	0	85.81	100.00
χ^2 dependency	380	50	0	**88.37**	100.00
# adjacency (*)	373	57	0	86.74	100.00
# adjacency (**)	358	72	0	83.26	100.00
# adjacency (***)	334	88	8	79.15	98.14
# adjacency (*, rev.)	370	59	1	86.25	99.77
# adjacency (**, rev.)	367	62	1	85.55	99.77
# adjacency (***, rev.)	351	79	0	81.63	100.00
Concatenation adj.	370	47	13	88.73	96.98
Concatenation dep.	366	43	21	**89.49**	95.12
Concatenation triple	238	37	155	**86.55**	63.95
Inflection Variability	198	49	183	80.16	57.44
Swap first two words	90	18	322	83.33	25.12
Reorder	320	78	32	80.40	92.56
Abbreviations	133	23	274	**85.25**	36.27
Possessives	48	7	375	**87.27**	12.79
Paraphrases	383	44	3	**89.70**	99.30
Surface features (sum)	382	48	0	**88.84**	100.00
Majority vote	403	17	10	95.95	97.67
Majority vote → right	*410*	*20*	*0*	***95.35***	***100.00***
Baseline (choose left)	361	69	0	83.95	100.00

Table 2: **Biomedical Set.**

as well, by returning too many false positives, but they work very well with concatenations and dashes: e.g., *T cell* is often written as *Tcell*.

As Table 4 shows, most of the surface features that we predicted to be right-bracketing actually indicated left. Overall, the surface features were very good at predicting left bracketing, but unreliable for right-bracketed examples. This is probably in part due to the fact that they look for adjacent words, i.e., they act as a kind of adjacency model.

We obtained our best overall results by combining the most reliable models, marked in bold in Tables 1, 2 and 4. As they have independent errors, we used a majority vote combination.

Table 3 compares our results to those of Lauer (1995) and of Lapata and Keller (2004). It is important to note though, that our results are *directly* comparable to those of Lauer, while the Keller&Lapata's are not, since they used half of the Lauer set for development and the other half for testing.[12] We, following Lauer, used everything for testing. Lapata & Keller also used the AltaVista search engine, which no longer exists in its earlier form. The table does not contain the results of Girju et al. (2005), who achieved 83.10% accuracy, but used a *supervised* algorithm and targeted bracketing *in context*. They further "shuffled" the Lauer's set, mixing it with additional data, thus making their results even harder to compare to these in the table.

Note that using page hits as a proxy for n-gram frequencies can produce some counter-intuitive 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 and Lapata (2003) for more issues.

[12] In fact, the differences are negligible; their system achieves pretty much the same result on the half split as well as on the whole set (personal communication).

Model	Acc. %
LEFT (baseline)	66.80
Lauer adjacency	68.90
Lauer dependency	77.50
Our χ^2 dependency	*79.92*
Lauer tuned	80.70
"Upper bound" (humans - Lauer)	81.50
Our majority vote → left	***89.34***
Keller&Lapata: LEFT (baseline)	63.93
Keller&Lapata: best BNC	68.03
Keller&Lapata: best AltaVista	78.68

Table 3: **Comparison to previous unsupervised results on *Lauer's set.*** The results of Keller & Lapata are on half of Lauer's set and thus are only indirectly comparable (note the different baseline).

5 Conclusions and Future Work

We have extended and improved upon the state-of-the-art approaches to NC bracketing using an unsupervised method that is more robust than Lauer (1995) and more accurate than Lapata and Keller (2004). Future work will include testing on NCs consisting of more than 3 nouns, recognizing the ambiguous cases, and bracketing NPs that include determiners and modifiers. We plan to test this approach on other important NLP problems.

As mentioned above, NC bracketing should be helpful for semantic interpretation. Another possible application is the refinement of parser output. Currently, NPs in the Penn TreeBank are flat, without internal structure. Absent any other information, probabilistic parsers typically assume right bracketing, which is incorrect about 2/3rds of the time for 3-noun NCs. It may be useful to augment the Penn TreeBank with dependencies inside the currently flat NPs, which may improve their performance overall.

Acknowledgements We thank Dan Klein, Frank Keller and Mirella Lapata for valuable comments, Janice Hamer for the annotations, and Mark Lauer for his dataset. This research was supported by NSF DBI-0317510, and a gift from Genentech.

Example	Predicts	Accuracy	Coverage
brain-stem cells	left	**88.22**	**92.79**
brain stem's cells	left	**91.43**	**16.28**
(brain stem) cells	left	96.55	6.74
brain stem (cells)	left	100.00	1.63
brain stem, cells	left	96.13	42.09
brain stem: cells	left	97.53	18.84
brain stem cells-death	left	80.69	60.23
brain stem cells/tissues	left	83.59	45.35
brain stem Cells	left	**90.32**	**36.04**
brain stem/cells	left	100.00	7.21
brain. stem cells	left	97.58	38.37
brain stem-cells	*right*	***25.35***	***50.47***
brain's stem cells	right	**55.88**	**7.90**
(brain) stem cells	*right*	*46.67*	*3.49*
brain (stem cells)	*right*	*0.00*	*0.23*
brain, stem cells	right	54.84	14.42
brain: stem cells	*right*	*44.44*	*6.28*
rat-brain stem cells	*right*	*17.97*	*68.60*
neural/brain stem cells	*right*	*16.36*	*51.16*
brain Stem cells	*right*	*24.69*	*18.84*
brain/stem cells	right	53.33	3.49
brain stem. cells	*right*	*39.34*	*14.19*

Table 4: **Surface features analysis (%s)**, run over the biomedical set.

References

Pamela Downing. 1977. On the creation and use of english compound nouns. *Language*, (53):810–842.

R. Girju, D. Moldovan, M. Tatu, and D. Antohe. 2005. On the semantics of noun compounds. *Journal of Computer Speech and Language - Special Issue on Multiword Expressions*.

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

Mirella Lapata and Frank Keller. 2004. The Web as a baseline: Evaluating the performance of unsupervised Web-based models for a range of NLP tasks. In *Proceedings of HLT-NAACL*, pages 121–128, Boston.

Mark Lauer. 1995. *Designing Statistical Language Learners: Experiments on Noun Compounds*. Ph.D. thesis, Department of Computing Macquarie University NSW 2109 Australia.

Mitchell Marcus. 1980. *A Theory of Syntactic Recognition for Natural Language*. MIT Press.

Preslav Nakov, Ariel Schwartz, Brian Wolf, and Marti Hearst. 2005. Scaling up BioNLP: Application of a text annotation architecture to noun compound bracketing. In *Proceedings of SIG BioLINK*.

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

Philip Resnik. 1993. *Selection and information: a class-based approach to lexical relationships*. Ph.D. thesis, University of Pennsylvania, UMI Order No. GAX94-13894.

Beatrice Warren. 1978. Semantic patterns of noun-noun compounds. In *Gothenburg Studies in English 41, Goteburg, Acta Universtatis Gothoburgensis*.

Y. Yang and J. Pedersen. 1997. A comparative study on feature selection in text categorization. In *Proceedings of ICML'97)*, pages 412–420.