The Automatic Acquisition of Frequencies of Verb Subcategorization Frames from Tagged Corpora

Akira Ushioda, David A. Evans, Ted Gibson, Alex Waibel

Computational Linguistics Program
Carnegie Mellon University
Pittsburgh, PA 15213-3890
aushioda@lcl.cmu.edu

Abstract

We describe a mechanism for automatically acquiring verb subcategorization frames and their frequencies in a large corpus. A tagged corpus is first partially parsed to identify noun phrases and then a linear grammar is used to estimate the appropriate subcategorization frame for each verb token in the corpus. In an experiment involving the identification of six fixed subcategorization frames, our current system showed more than 80% accuracy. In addition, a new statistical approach substantially improves the accuracy of the frequency estimation.

1 Introduction

When we construct a grammar, there is always a trade-off between the coverage of the grammar and the ambiguity of the grammar. If we hope to develop an efficient high-coverage parser for unrestricted texts, we must have some means of dealing with the combinatorial explosion of syntactic ambiguities. While a general probabilistic optimization technique such as the Inside-Outside algorithm ([Baker, 1979], [Lauri and Young, 1990], [Jelinek *et al.*, 1990], [Carroll and Charniak, 1992]) can be used to reduce ambiguity by providing estimates on the applicability of the context-free rules in a grammar (for example), the algorithm does not take advantage of lexical information, including such information as verb subcategorization frame preferences. Discovering or acquiring lexically-sensitive linguistic structures from large corpora may offer an essential complementary approach.

Verb subcategorization (verb-subcat) frames represent one of the most important elements of grammatical/lexical knowledge for efficient and reliable parsing. At this stage in the computational-linguistic exploration of corpora, dictionaries are still probably more reliable than automatic acquisition systems as a source of subcategorization (subcat) frames for verbs. The Oxford Advanced Learners Dictionary (OALD) [Hornby, 1989], for example, uses 32 verb patterns to describe a usage of each verb for each meaning of the verb. However, dictionaries do not provide quantitative information such as how often each verb is used with each of the possible subcat frames. Since dictionaries are repositories, primarily, of what is possible, not what is most likely, they tend to contain information about rare usage [de Marken, 1992]. But without information about the frequencies of the subcat frames we find in dictionaries, we face the prospect of having to treat each frame as equiprobable in parsing. This can lead to serious inefficiency. We also know that the frequency of subcat frames can vary by domain; frames that are very rare in one domain can be quite common in another. If we could automatically determine the frequencies of subcat frames for domains, we would be able to tailor parsing with domain-specific

heuristics. Indeed, it would be desirable to have a subcat dictionary for each possible domain.

This paper describes a mechanism for automatically acquiring subcat frames and their frequencies based on a tagged corpus. The method utilizes a tagged corpus because (i) we don't have to deal with a lexical ambiguity (ii) tagged corpora in various domains are becoming readily available and (iii) simple and robust tagging techniques using such corpora recently have been developed ([Church, 1988], [Brill, 1992]).

Brent reports a method for automatically acquiring subcat frames but without frequency measurements ([Brent and Berwick, 1991], [Brent, 1991]). His approach is to count occurrences of those unambiguous verb phrases that contain no noun phrases other than pronouns or proper nouns. By thus restricting the "features" that trigger identification of a verb phrase, he avoids possible errors due to syntactic ambiguity. Although the rate of false positives is very low in his system, his syntactic features are so selective that most verb tokens fail to satisfy them. (For example, verbs that occurred fewer than 20 times in the corpus tend to have no co-occurrences with the features.) Therefore his approach is not useful in determining verb-subcat frame frequencies.

To measure frequencies, we need, ideally, to identify a subcat frame for each verb token in the corpus. This, in turn, requires a full parse of the corpus. Since manually parsed corpora are rare and typically small, and since automatically parsed corpora contain many errors (given current parsing technologies), an alternative source of useful linguistic structure is needed. We have elected to use partially parsed sentences automatically derived from a lexically-tagged corpus. The partial parse contains information about minimal noun phrases (without PP attachment or clausal complements). While such derived information about linguistic structure is less accurate and complete than that available in certified, hand-parsed corpora, the approach promises to generalize and to yield large sample sizes. In particular, we can use partially parsed corpora to measure verb-subcat frame frequencies.

2 Method

The procedure to find verb-subcat frequencies, automatically, is as follows.

(1) Make a list of verbs out of the tagged corpus.

(2) For each verb on the list (the "target verb"),

  (2.1) Tokenize each sentence containing the target verb in the following way:
  All the noun phrases except pronouns are tokenized as "n" by a noun phrase parser and all the rest of the words are also tokenized following the schmema in Table 1. For example, the sentence "The corresponding mental-state verbs do not follow [target verb] these rules in a straightforward way" is transformed to a sequence of tokens "bnvaknpne".

  (2.2) Apply a set of subcat extraction rules to the tokenized sentences. These rules are written as regular expressions and they are obtained through the examination of occurrences of a small sample of verbs in a training text.

Note that in the actual implementation of the procedure, all of the redundant operations are eliminated. Our NP parser also uses a finite-state grammar. It is designed

b:	sentence initial maker	e:	sentence final maker
k:	target verb	t:	"to"
i:	pronoun	m:	modal
n:	noun phrase	w:	relative pronoun
v:	finite verb	a:	adverb
u:	participial verb	x:	punctuation
d:	base form verb	c:	complementizer "that"
p:	preposition	s:	the rest

Table 1: List of Symbols/Categories

especially to support identification of verb-subcat frames. One of its special features is that it detects time-adjuncts such as "yesterday", "two months ago", or "the following day", and eliminates them in the tokenization process. For example, the sentence "He told the reporters the following day that..." is tokenized to "bivnc..." instead of "bivnnc...".

3 Experiment on Wall Street Journal Corpus

We used the above method in experiments involving a tagged corpus of Wall Street Journal (WSJ) articles, provided by the Penn Treebank project. Our experiment was limited in two senses. First, we treated all prepositional phrases as adjuncts. (It is generally difficult to distinguish complement and adjunct PPs.) Second, we measured the frequencies of only six fixed subcat frames for verbs in non-participle form. (This does not represent an essential shortcoming in the method; we only need to have additional subcat frame extraction rules to accommodate participles.)

We extracted two sets of tagged sentences from the WSJ corpus, each representing 3-MBytes and approximately 300,000 words of text. One set was used as a training corpus, the other as a test corpus. Table 2 gives the list of verb-subcat frame extraction rules obtained (via examination) for four verbs "expect", "reflect", "tell", and "give", as they occurred in the training corpus. Sample sentences that can be captured by each set of rules are attached to the list. Table 3 shows the result of the hand comparison of the automatically identified verb-subcat frames for "give" and "expect" in the test corpus. The tabular columns give actual frequencies for each verb-subcat frame based on manual review and the tabular rows give the frequencies as determined automatically by the system. The count of each cell ($[i, j]$) gives the number of occurrences of the verb that are assigned the i-th subcat frame by the system and assigned the j-th frame by manual review. The frame/column labeled "REST" represents all other subcat frames, encompassing such subcat frames as those involving wh-clauses, verb-particle combinations (such as "give up"), and no complements.

Despite the simplicity of the rules, the frequencies for subcat frames determined under automatic processing are very close to the real distributions. Most of the errors are attributable to errors in the noun phrase parser. For example, 10 out of the 13 errors in the [NP,NP+NP] cell under "give" are due to noun phrase parsing errors such as the misidentification of a N–N sequence (e.g., *"give [NP government officials rights] against the press" vs. "give [NP government officials] [NP rights] against the press").

	Frame	Rule
1.	NP+NP	k(i|n)n
2.	NP+CL	k(i|n(pn)*)c
		k(i|n)(i|n)a*(m|v)
3.	NP+INF	k(i|n(pn)*)ta*d
4.	CL	kc
		k(i|n)a*(m|v)
5.	NP	k(i|n)/[^mvd]
		#pw(i|n(pn)*)a*m?a*k/[^t]
6.	INF	kta*d

Notes:

NP: noun phrase
CL: that-clause with and without the complementizer "that"
INF: "to" + infinitive

x* matches a sequence of any number of x's including zero x
x? is either x or empty
(x|y) matches either x or y
[^xyz] matches any token except x, y, and z
#x(sequence) matches (sequence) that is not directly preceded by x
x/y matches x if x is immediately followed by y

Sample Sentences:

Frame 1. "...gives current management enough time to work on..."

Frame 2. "...tell the people in the hall that..."; "...told him the man would..."

Frame 3. "...expected the impact from the restructuring to make..."

Frame 4. "...think that..."; "...thought the company eventually responded..."

Frame 5. "...saw the man..."; "...which the president of the company wanted..."
but not
"...saw him swim..."; "...(hotel) in which he stayed..."; "...(gift) which he expected to get..."

Frame 6. "...expects to gain..."

Table 2: Set of Subcategorization Frame Extraction Rules

"Give"

Real Occurrences

Output of System	NP+NP	NP+CL	NP+INF	NP	CL	INF	REST	Total
NP+NP	52	0	0	0	0	0	0	52
NP+CL	1	0	0	0	0	0	0	1
NP+INF	2	0	0	0	0	0	0	2
NP	13	0	0	27	0	0	0	40
CL	0	0	0	0	0	0	0	0
INF	0	0	0	0	0	0	0	0
REST	1	0	0	4	0	0	9	14
Total	69	0	0	31	0	0	9	109

"Expect"

Real Occurrences

Output of System	NP+NP	NP+CL	NP+INF	NP	CL	INF	REST	Total
NP+NP	0	0	0	0	0	0	0	0
NP+CL	0	0	0	0	0	0	0	0
NP+INF	0	0	55	1	0	0	0	56
NP	0	0	4	28	0	0	0	32
CL	0	0	0	0	8	0	0	8
INF	0	0	0	0	0	40	0	40
REST	0	0	1	6	0	0	7	14
Total	0	0	60	35	8	40	7	150

Table 3: Subcategorization Frame Frequencies

acquire	end	like	spend
build	expand	need	total
close	fail	produce	try
comment	file	prove	use
consider	follow	reach	want
continue	get	receive	work
design	help	reduce	
develop	hold	see	
elect	let	sign	

Table 4: Verbs Tested

THIS PAGE INTENTIONALLY LEFT BLANK

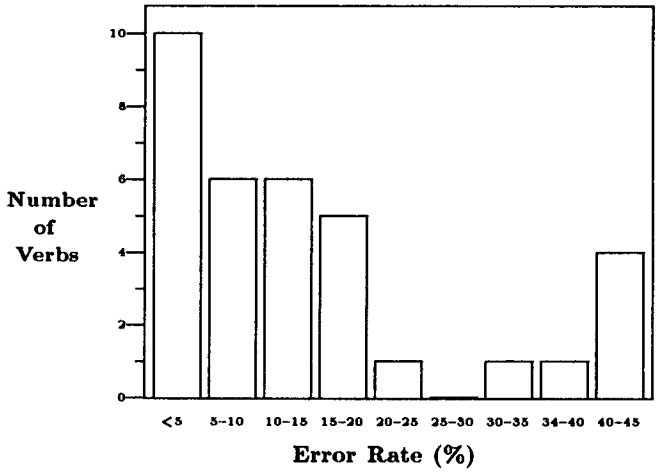

Figure 1: Distribution of Errors

To measure the total accuracy of the system, we randomly chose 33 verbs from the 300 most frequent verbs in the test corpus (given in Table 4), automatically estimated the subcat frames for each occurrence of these verbs in the test corpus, and compared the results to manually determined subcat frames.

The overall results are quite promising. The total number of occurrences of the 33 verbs in the test corpus (excluding participle forms) is 2,242. Of these, 1,933 were assigned correct subcat frames by the system. (The 'correct'-assignment counts always appear in the diagonal cells in a comparison table such as in Table 3.) This indicates an overall accuracy for the method of 86%.

If we exclude the subcat frame "REST" from our statistics, the total number of occurrences of the 33 verbs in one of the six subcat frames is 1,565. Of these, 1,311 were assigned correct subcat frames by the system. This represents 83% accuracy.

For 30 of the 33 verbs, both the first and the second (if any) most frequent subcat frames as determined by the system were correct. For all of the verbs except one ("need"), the most frequent frame was correct.

Figure 1 is a histogram showing the number of verbs within each error-rate zone. In computing the error rate, we divide the total 'off-diagonal'-cell counts, excluding the counts in the "REST" column, by the total cell counts, again excluding the "REST" column margin. Thus, the off-diagonal cell counts in the "REST" row, representing instances where one of the six actual subcat frames was misidentified as "REST", are counted as errors. This formula, in general, gives higher error rates than would result from simply dividing the off-diagonal cell counts by the total cell counts.

Overall, the most frequent source of errors, again, was errors in noun phrase boundary detection. The second most frequent source was misidentification of infinitival 'purpose' clauses, as in "he used a crowbar to open the door". "To open the door" is a 'purpose' adjunct modifying either the verb phrase "used a crowbar" or the main clause "he used a crowbar". But such adjuncts are incorrectly judged to be complements of their main verbs

by the subcat frame extraction rules in Table 2. In formulating the rules, we assumed that a 'purpose' adjunct appears effectively randomly and much less frequently than infinitival complements. This is true for our corpus in general; but some verbs, such as "use" and "need", appear relatively frequently with 'purpose' infinitivals. In addition to errors from parsing and 'purpose' infinitives, we observed several other, less frequent types of errors. These, too, pattern with specific verbs and do not occur randomly across verbs.

4 Statistical Analysis

For most of the verbs in the experiment, our method provides a good measure of subcat frame frequencies. However, some of the verbs seem to appear in syntactic structures that cannot be captured by our inventory of subcat frames. For example, "need" is frequently used in relative clauses without relative pronouns, as in "the last thing they need". Since this kind of relative clauses cannot be captured by the rules in Table 2, each occurrence of these relative clause causes an error in measurement. It is likely that there are many other classes of verbs with distinctive syntactic preferences. If we try to add rules for each such class, it will become increasingly difficult to write rules that affect only the target class and to eliminate undesirable rule interactions.

In the following sections, we describe a statistical method which, based on a set of training samples, enables the system to learn patterns of errors and substantially increase the accuracy of estimated verb-subcat frequencies.

4.1 General Scheme

The method described in Section 2 is wholly deterministic; it depends only on one set of subcat extraction rules which serve as filters. Instead of treating the system output for each verb token as an estimated subcat frame, we can think of the output as one feature associated with the occurrence of the verb. This single feature can be combined, statistically, with other features in the corpus to yield more accurate characterizations of verb contexts and more accurate subcat-frame frequency estimates. If the other features are capturable via regular-expression rules, they can also be automatically detected in the manner described in the Section 2. For example, main verbs in relative clauses without relative pronouns may have a higher probability of having the feature "nnk", i.e., "(NP)(NP)(VERB)".

More formally, let Y be a response variable taking as its value a subcat frame. Let $X_1, X_2, \ldots, X_N$ be explanatory variables. Each X_i is associated with a feature expressed by one or a set of regular expressions. If a feature is expressed by one regular expression (R), the value of the feature is 1 if the occurrence of the verb matches R and 0 otherwise. If the feature is expressed by a set of regular expressions, its value is the label of the regular expression that the occurrence of the verb matches. The set of regular expressions in Table 2 can therefore be considered to characterize one explanatory variable whose value ranges from (NP+NP) to (REST).

Now, we assume that a training corpus is available in which all verb tokens are given along with their subcat frames. By running our system on the training corpus, we can automatically generate a $(N+1)$-dimensional contingency table. Table 3 is an example of a 2-dimensional contingency table with X = <OUTPUT OF SYSTEM> and Y = <REAL OCCURRENCES>. Using loglinear models [Agresti, 1990], we can derive fitted values of

each cell in the $(N+1)$-dimensional contingency table. In the case of a saturated model, in which all kinds of interaction of variables up to $(N+1)$-way interactions are included, the raw cell counts are the Maximum Likelihood solution. The fitted values are then used to estimate the subcat frame frequencies of a new corpus as follows.

First, the system is run on the new corpus to obtain an N-dimensional contingency table. This table is considered to be an $X_1 - X_2 - \cdots - X_N$-marginal table. What we are aiming at is the Y margins that represent the real subcat frame frequencies of the new corpus. Assuming that the training corpus and the new corpus are homogeneous (e.g., reflecting similar sub-domains or samples of a common domain), we estimate the Y margins using Bayes theorem on the fitted values of the training corpus as follows:

$$E(Y = k \mid X_1 - X_2 - \cdots - X_N \textit{ marginal table of the new corpus})$$

$$= \sum_{i_1}\sum_{i_2}\cdots\sum_{i_N} \mathcal{N}_{i_1 i_2 \cdots i_N +}\, P(Y = k \mid X_1 = i_1, X_2 = i_2, \cdots, X_N = i_N)$$

$$= \sum_{i_1}\sum_{i_2}\cdots\sum_{i_N} \mathcal{N}_{i_1 i_2 \cdots i_N +} \frac{P(X_1 = i_1, X_2 = i_2, \cdots, X_N = i_N \mid Y = k)\, P(Y = k)}{\sum_{k'}[P(X_1 = i_1, X_2 = i_2, \cdots, X_N = i_N \mid Y = k')\, P(Y = k')]}$$

$$= \sum_{i_1}\sum_{i_2}\cdots\sum_{i_N} \mathcal{N}_{i_1 i_2 \cdots i_N +} \frac{\mathcal{M}_{i_1 i_2 \cdots i_N k}}{\sum_{k'} \mathcal{M}_{i_1 i_2 \cdots i_N k'}}$$

where $\mathcal{N}_{i_1 i_2 \cdots i_n +}$ is the cell count of the $X_1 - X_2 - \cdots - X_N$ marginal table of the new corpus obtained as the system output, and $\mathcal{M}_{i_1 i_2 \cdots i_N k}$ is the fitted value of the $(N+1)$-dimensional contingency table of the training corpus based on a particular loglinear model.

4.2 Lexical Heuristics

The simplest application of the above method is to use a 2-way contingency table, as in Table 3. There are two possibilities to explore in constructing a 2-way contingency table. One is to sum up the cell counts of all the verbs in the training corpus and produce a single (large) general table. The other is to construct a table for each verb. Obviously the former approach is preferable if it works. Unfortunately, such a table is typically too general to be useful; the estimated frequencies based on it are less accurate than raw system output. This is because the sources of errors, viz., the distribution of off-diagonal cell counts of 2-way contingency tables, differ considerably from verb to verb. The latter approach is problematic if we have to make such a table for each domain. However, if we have a training corpus in one domain, and if the heuristics for each verb extracted from the training corpus are also applicable to other domains, the approach may work.

To test the latter possibility, we constructed a contingency table for the verb from the test corpus described in the Section 3 that was most problematic (least accurately estimated) among the 33 verbs—"need". Note that we are using the test corpus described in the Section 3 as a training corpus here, because we already know both the measured frequency and the hand-judged frequency of "need" which are necessary to construct a contingency table. The total occurrence of this verb was 75. To smooth the table, 0.1 is added to all the cell counts. As new test corpora, we extracted another 300,000 words of tagged text from the WSJ corpus (labeled "W3") and also three sets of 300,000 words of tagged text from the Brown corpus (labeled "B1", "B2", and "B3"), as retagged under the

W3	NP+NP	NP+CL	NP+INF	NP	CL	INF	REST
Measured	2.4	0.0	10.6	44.7	1.2	31.8	9.4
By Hand	0.0	0.0	0.0	69.4	0.0	30.6	0.0
Estimated	0.0	0.0	0.0	66.3	0.0	30.1	3.6
			Total Occurrences: 85				

B1	NP+NP	NP+CL	NP+INF	NP	CL	INF	REST
Measured	1.8	0.9	7.9	38.6	1.8	14.9	34.2
By Hand	0.0	0.0	0.0	72.8	0.0	15.8	11.4
Estimated	0.0	0.0	0.0	76.6	0.0	14.4	9.1
			Total Occurrences: 114				

B2	NP+NP	NP+CL	NP+INF	NP	CL	INF	REST
Measured	0.0	1.4	8.7	40.6	1.4	17.4	30.4
By Hand	0.0	0.0	0.0	73.9	0.0	18.8	7.2
Estimated	0.0	0.0	0.0	76.1	0.0	16.4	7.5
			Total Occurrences: 69				

B3	NP+NP	NP+CL	NP+INF	NP	CL	INF	REST
Measured	3.3	0.0	1.7	30.0	3.3	31.7	30.0
By Hand	0.0	0.0	0.0	60.0	0.0	28.3	11.7
Estimated	0.0	0.0	0.0	61.4	0.0	29.8	8.8
			Total Occurrences: 60				

Table 5: Statistical Estimation (Unit = %) for the Verb "Need"

Penn Treebank tagset. All the training and test corpora were reviewed—and judged—by hand.

Table 5 gives the frequency distributions based on the system output, hand judgement, and statistical analysis. (As before, we take the hand judgement to be the gold standard, the actual frequency of a particular frame.) After the Y margins are statistically estimated, the least estimated Y values less than 1.0 are truncated to 0. (These are considered to have appeared due to the smoothing.)

In all of the test corpora, the method gives very accurate frequency distribution estimates. Big gaps between the automatically-measured and manually-determined frequencies of "NP" and "REST" are shown to be substantially reduced through the use of statistical estimation. This result is especially encouraging because the heuristics obtained in one domain are shown to be applicable to a considerably different domain. Furthermore, by combining more feature sets and making use of multi-dimensional analysis, we can expect to obtain more accurate estimations.

5 Conclusion and Future Direction

We have demonstrated that by combining syntactic and statistical analysis, the frequencies of verb-subcat frames can be estimated with high accuracy. Although the present system measures the frequencies of only six subcat frames, the method is general enough to be extended to many more frames. The traditional application of regular expressions as rules for deterministic processing has self-evident limitations since a linear grammar is not powerful enough to capture general linguistic phenomena. The statistical method we propose uses regular expressions as filters for detecting specific features of the occurrences of verbs and employs multi-dimensional analysis of the features based on loglinear models and Bayes Theorem.

We expect that by identifying other useful syntactic features we can further improve the accuracy of the frequency estimation. Such features can be regarded as characterizing the syntactic context of the verbs, quite broadly. The features need not be linked to a local verb context. For example, a regular expression such as "`w[^vex]*k`" can be used to find cases where the target verb is preceded by a relative pronoun such that there is no other finite verb or punctuation or sentence final period between the relative pronoun and the target verb.

If the syntactic structure of a sentence can be predicted using only syntactic and lexical knowledge, we can hope to estimate the subcat frame of each occurrence of a verb using the context expressed by a set of features. We thus can aim to extend and refine this method for use with general probabilistic parsing of unrestricted text.

6 Acknowledgements

We thank Teddy Seidenfeld, Jeremy York, and Alex Franz for their comments and discussions with us. We remain, of course, solely responsible for any errors or inadequacies in the paper.

References

[Agresti, 1990] A. Agresti. *Categorical Data Analysis*. New York, NY: John Wiley and Sons, 1990.

[Baker, 1979] J. Baker. "Trainable Grammars for Speech Recognition". In D.H. Klatt and J.J. Wolf (eds.), *Speech Communication Papers for the 97th Meeting of the Acoustic Society of America*, 1979, pp. 547–550.

[Brent, 1991] M.R. Brent. "Automatic Acquisition of Subcategorization Frames from Untagged Text". *Proceedings of the 29th Annual Meeting of the ACL*, 1991.

[Brent and Berwick, 1991] M.R. Brent and R.C. Berwick. "Automatic Acquisition of Subcategorization Frames from Tagged Text". In *Proceedings of the DARPA Speech and Natural Language Workshop*, Morgan Kaufmann, 1991.

[Brill, 1992] E. Brill. "A Simple Rule-Based Part of Speech Tagger". In *Proceedings of the DARPA Speech and Natural Language Workshop*, Morgan Kaufmann, 1992.

[Carroll and Charniak, 1992] G. Carroll and E. Charniak. "Learning Probabilistic Dependency Grammars from Labelled Text". In *Working Notes of the Symposium on Probabilistic Approaches to Natural Language*, AAAI Fall Symposium Series, 1992.

[Church, 1988] K.W. Church. "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text". In *Proceedings of the Second Conference on Applied Natural Language Processing*, 1988.

[de Marken, 1992] C.G. de Marcken. "Parsing the LOB Corpus". In *Proceedings of the 28th Annual Meeting of the ACL*, 1990, pp. 243–251.

[Hornby, 1989] A.S. Hornby, (ed.). *Oxford Advanced Learner's Dictionary of Current English*. Oxford, UK: Oxford University Press, 1989.

[Jelinek *et al.*, 1990] F. Jelinek, L.D. Lafferty, and R.L. Mercer. *Basic Method of Probabilistic Context Free Grammars*. Technical Report RC 16374 (72684), IBM, Yorktown Heights, NY 10598, 1990.

[Lauri and Young, 1990] K. Lari and S.J. Young. "The Estimation of Stochastic Context-Free Grammars Using the Inside-Outside Algorithm". *Computer Speech and Language*, *4*, 1990, pp. 35–56.