CONSTRUCTING A PHRASE STRUCTURE GRAMMAR BY INCORPORATING LINGUISTIC KNOWLEDGE AND STATISTICAL LOG-LIKELIHOOD RATIO

Keh-Yih Su*, Yu-Ling Hsu**, and Claire Saillard***

*Department of Electrical Engineering National Tsing Hua University Hsinchu, Taiwan, R.O.C. email: kysu@ee.nthu.edu.tw **Behavior Design Corporation 2nd. Fl., 28 R&D Road II Science-Based Industrial Park Hsinchu, Talwan, R.O.C. *** Institute of Linguistics National Tsing Hua University Hsinchu, Taiwan, R.O.C.

ABSTRACT

Phrase structure grammar is one of the most important components in a syntax-oriented parsing system. However, constructing an adequate PSG is an arduous task. Either the traditional linguistic approach or the fully automatic inference approach has encountered several difficulties.

Thus, a human-machine cooperative method is suggested in this paper as a better approach. A statistical tool, **Log-Likelihood Ratio**, is proposed to enhance the productivity of human grammar writers. The Log-Likelihood Ratio of co-occurring tags is automatically computed by the computer to indicate the strength of linear association. The task of linguists is then to verify the relevance of groupings based on their linguistic knowledge. The advantages of this approach over other methods are pointed out, and the actual procedures are illustrated by a pilot experiment of constructing a Mandarin PSG. The experimental result shows the feasibility of the proposed approach.

1. Introduction

In a syntax oriented parsing system, parsing usually amounts to consulting a phrase structure grammar (PSG, hereafter) to check the well-formedness of the input strings and to generate their corresponding syntactic structures accordingly. Thus, PSG is one of the most important components of the whole parsing system.

However, constructing an adequate PSG is an arduous task. Traditional approaches resort only to linguists' own knowledge, and therefore are extremely labor-intensive and prone to incompleteness and incoherence in practical large-scale systems. Recently, owing to the advance of computer technology in providing cheap and fast computational power and the increasing availability of machine-readable corpora, corpus-based statistical approaches are gaining prevalence in the community of computational linguistics. Plenty of systems propose to use statistics in their researches, including lexical analysis, category disambiguation, semantic models etc.¹ However, as for PSG construction, there are still no satisfactory methods available, neither traditional nor statistical.

Thus, a statistical tool, Log-Likelihood Ratio, is proposed in this paper to provide clues for linear association and enhance the productivity of human grammar writers. This approach intends to incorporate statistical information and linguistic knowledge in order to benefit from both the simple, objective, consistent characteristics of statistics and the better deductive power of ready-made linguistic analyses.

In the next section, two previous approaches of constructing a PSG are presented. Their advantages and drawbacks are demonstrated in detail. Then the Log-Likelihood-Ratio statistic is introduced. A fully automatic approach based on a so-called Generalized Mutual Information will also be discussed, with its shortcomings. Finally, the proposed cooperative approach will be presented. The actual procedures will be illustrated by a pilot experiment of constructing a Mandarin PSG. The experimental result shows the feasibility of the proposed approach.

2. Previous Approaches

In this section, we will describe two extremely different approaches of constructing a PSG. Their advantages and drawbacks will be discussed in detail.

2.1 Relying on Linguistic Knowledge Only

This approach has been traditionally used as a dominant way for constructing a PSG. In the initial phase, the cost of this method is minimal, because no collection of a large machine-readable database or preprocessing of the database is needed. A few linguists can build a preliminary PSG of a language in a relatively short time, by incorporating the ready-made theoretical linguistic analyses of this language. Since well-known linguistic analyses have gone through rigorous argumentations and been well-tested by lots of empirical data, they provide much insight about the language and their descriptive power is relatively strong.

However, theoretical linguistic researchers are apt to focus their attention on theoretically interesting phenomena and sweep the residual problems under the carpet. Unfortunately, theoretically interesting phenomena do not necessarily correlate to frequently occurring phenomena in real texts. Thus, although many aspects of grammatical structure are well-known and uncontroversial, authentic material still includes massive amounts of phenomena which have been

¹ See [9], [12], and [16], etc.

ignored or have not yet received consentient linguistic analyses. Especially in a language like Mandarin Chinese, where the linguistic phenomena are poorly studied, the contribution of the ready-made linguistic analyses to the construction of a Mandarin PSG is even more limited.

If ready-made linguistic analyses offer no guidance to the construction of a PSG, linguists have to work based on their own linguistic knowledge. In most cases, linguists start with a small set of data which is the basis for the first formulation of the grammar. Then, they gradually expand the data under consideration, using new data to test their original hypothesis and make decisions among competing analyses. The grammar is under reformulation until it covers most of the sentences in consideration. This method works well in theoretical researches or for small scale systems. However, when the set of data has been enlarged to thousands or millions of sentences, human simply can no longer successfully handle all the trivial linguistic phenomena, let alone the complicated interrelations among rules. Consequently, a purely linguistic approach to grammar construction arouses several problems in large scale systems.

Firstly, the PSG constructed in this way is prone to errors of omission. Human is not good at managing massive amounts of data. Without the help of the computer, linguists may ignore many trivial phenomena they do want to cover, and occasional mistakes are also inevitable.

Secondly, no simple and objective measure of the data is available for linguists to make tradeoffs between the coverage and the efficiency of the PSG. Ideally, a good PSG should define the class of "all and only" well-formed sentences of the language. But since authentic language is much more complex than theoretical linguists' descriptions commonly imply, this goal is hard to be achieved in practical systems. It may be clearer from what Sampson says : " If the activity of revising a generative grammar in response to recalcitrant authentic examples were ever to terminate in a perfectly leak-free grammar, that grammar would surely be massively more complicated than any extant grammar, and would thus pose correspondingly massive problems with respect to incorporation into a system of automatic analysis."² That is to say, attempting to construct a grammar accounting for all constructions in real-life texts is not feasible. Thus, some "omissions" of data are required. Most practical NLP systems will define the subject domain and style for their input texts and evaluate the importance of each construction according to the frequency of its real occurrences. If certain constructions have few occurrences in their domain, they will be discarded to avoid causing extra-complication of the system. However, without statistical information as a reference, the tradeoffs are difficult to be made.

Thirdly, linguists in this way do not have a general view of the linguistic phenomena involved during the process of grammar construction, and therefore modification shall very likely have to be made on preceding decisions if new data triggers new arguments in favor of a different solution. But without an objective measure of the real coverage of the PSG, grammar

² See [12], Chap. 2, 20.

-

writers cannot predict the actual influences caused by the modification of rules. Back-and-forth modifications are therefore hard to be avoided because linguists cannot guarantee their successive alterations of the rules lead to a global enhancement of the whole system. Consequently, the revision process will be full of back-and-forth operations, and it would be hard to imagine how the process should ever be concluded.

2.2 A Fully Automatic Approach Based on Grammatical Inference

Opposed to the purely linguistic approach mentioned above, a fully automatic approach based on grammatical inference has also been proposed. The principle of grammatical inference is to extract a grammar from a set of sentences, i.e. the sample or the learning set, which generates a set of sentences containing the sample. This procedure is an important subject in the study of syntactic pattern recognition because of its automatic learning capability. Several algorithms have been proposed and discussed.³ Potential engineering applications of grammatical inference include areas of information retrieval, translation and compiling, and artificial intelligence, etc.

Generally speaking, the inferred grammar is a set of rules for describing the given finite set of strings from L(G), the language generated by G, and predicting other strings which in some sense are of the same nature as the given set. A model for the inference of string grammars is shown in Figure 1. A set of sample terminal strings $\{x_i\}$ is fed into an adaptive learning algorithm, represented by the box in Figure 1, and a grammar G which is compatible with the given strings is obtained from the output.⁴



Figure 1 grammatical inference of string grammars

Thus, it is possible to directly infer a PSG from a set of sample sentences. In doing grammatical inference, the most popular method is to deduce the grammar in Chomsky Normal-Form. The Chomsky Normal-Form Theorem states that every context-free language can be generated by a grammar in which all productions are of the form $A \rightarrow BC$ or $A \rightarrow a$. Here A, B, and C are variables and a is a terminal. By the strategy of grammatical inference, a grammar with Chomsky Normal Form can be automatically inferred from the sample corpus.

At the first step, a set of sentences is selected as the corpus. Then, the corpus is tagged with lexical categories to reduce the number of terminal symbols of grammar rules. Finally,

³ Detailed discussions on grammatical inference can be found in [11].

⁴ See [11] and [17].

the inference algorithm is performed by the computer to automatically infer a grammar from the tagged corpus.

This automatic approach has some advantages. Firstly, since the inferred procedures are performed by the computer, it can be proved that the inferred grammar will perfectly cover all the sentences in the sample corpus.

Secondly, a fully automatic approach can reduce human intervention to a minimum. As humans are recognized as the most precious, yet most costly, resources in NLP systems, reducing human intervention will greatly enhance the cost-effectiveness of a system.

Nevertheless, this approach has several serious drawbacks. Firstly, since the the automatic construction of PSG does not take semantic relevance into consideration, the constituents constructed in this way are just ad hoc groupings which may not correspond to any traditional semantic concept. For many applications of NLP, such as text understanding and machine translation, semantic interpretation is an important process after syntactic parsing. Thus, the mismatch between the automatically-trained syntactic model and the traditional semantic model will cause difficulties for human linguists to attach semantic information to the syntactically analyzed structures.

Secondly, since the syntactic grammar inferred by automatical procedures is dramatically different from that of standard linguistic researches, the inferred grammar will not be able to couple with existing linguistic theories and thus to take advantage of the achievements of linguistic researches. As mentioned previously, most of the linguistic analyses are well-motivated and well-tested. They are valuable resources for related researches. Thus, it is a mistake to overlook the value of linguistic information and adopt a thoroughgoing automatic approach.

Thirdly, it is clear that the choice of the initial sample is critical in this approach. If the sample is too small, since all the rules are acquired exclusively from the corpus, the grammar may not be able to account for phenomena outside the sample space. But if the size of the sample is large, the number of inferred rules may become astronomically large and greatly increase the complexity of processing.

III. Using Log-Likelihood Ratio to Construct a PSG

As we have discussed, previous approaches for constructing a PSG have encountered several serious problems. Thus, a statistical tool, called Log-Likelihood Ratio, is proposed in this section to fertilize the construction of a PSG.

3.1 What Is Log-Likelihood Ratio

Log-Likelihood Ratio (LLR, hereafter) is a statistic measure of word associations. It compares the probability of a group of tags to occur together (joint probability) to their probability of occurring independently.

The bigram (with window size of 2) LLR, also called Mutual Information in the literatures, is computed by the formula:⁵

$$LLR_{2}(x, y) = I(x; y) = \log_{2} \frac{P(x, y)}{P(x) \times P(y)}$$

where x and y are two tags in the corpus, and $LLR_2(x, y)$ (or I(x, y)) is the bigram Log-Likelihood Ratio (or Mutual Information) of the two tags x and y (in this order). P(x) is evaluated as the relative frequency of the number of occurrences of x with respect to the number of total instances of singletons.

If there is a genuine association between x and y, then the joint probability P(x, y) will be much larger than the chance $P(x) \times P(y)$, and consequently $LLR_2(x, y) >> 0$. If there is no interesting relationship between x and y, then $P(x, y) = P(x) \times P(y)$, and thus $LLR_2(x, y) = 0$. If x and y are in complementary distribution, then P(x, y) will be much less than $P(x) \times P(y)$, and thus $LLR_2(x; y) \ll 0$.

3.2 An Automatic Approach Using Generalized Mutual Information

In the past few years, Mutual Information has been used in many areas of natural language processing, and has shown its success in different applications.⁶ Recently, based on so-called Generalized Mutual Information (GMI, hereafter), an automatic constituent boundary parsing algorithm has been developed, which can derive a syntactic (unlabelled) bracketing for input tagged texts. In this approach, the tag sequences are processed using an n-ary-branching recursive function which branches at the minimum GMI value of the given window. Besides, for exceptional cases, a distituent grammar is constructed to specify a list of tag pairs which cannot be adjacent within a constituent.

Unfortunately, this approach has some drawbacks. Firstly, the formula of the GMI is not theoretically well-supported. It is heuristically expressed as a weighted sum of the Mutual Information based on the substring of the given context.⁷

Secondly, as mentioned, the bracketing of sentences in this approach is majorily determined by the value of GMI. A local minimum suggests the place to bracket. But this way of constructing

⁵ For more details, readers are referred to [9].

⁶ For example, [6], [7], [8], [9], and [18] have shown that Mutual Information is helpful in their researches.

⁷ Interested Readers are referred to [2] for more details.

constituents still deviates from that of the standard linguistic researches. Conventionally, linguists determine the constituency of words not only by the strength of their linear co-occurrences, but more importantly also by their semantic relevance, or their substitutability and movability. It should be noted that tags in the same constituents should have higher GMI, but tags with higher GMI do not necessarily imply they are belonging to the same constituents. For example, verbs and determiners frequently occur together in sentences, thus the GMI for verb-determiner pair will be relatively high. However, linguists will never group verbs and determiners into the same constituent because they do not correspond to any semantic concept and do not act as a unit in syntactic operations (e.g. movement). Although the distituent grammar is constructed to make up this shortcoming, the adequate list of distituents is hard to be defined and is nevertheless source of inaccuracies.⁸ As a consequence, a cooperative approach which incorporate both linguistic knowledge and statistical information is proposed in this paper to construct a PSG.

3.3 A Cooperative Approach Combining Linguistic Knowledge and LLR

This approach combines the advantages of the conventional linguistic knowledge-based method and those of the corpus-based, statistical approach. Firstly, a corpus with lexical tags is still required. Secondly, the LLR of co-occurring tags is automatically computed by the computer. The task of linguists is then to decide whether the linearly highly associated tags are belonging to the same constituents, or to highly associated but distinct constituents. That is, the grouping of tags indicated by the computer is further confirmed by linguists' knowledge about the syntactic constituency.

On the one hand, the advantage of incorporating linguists' knowledge about constituency is to eliminate the drawbacks of the automatic construction of PSG, so as to couple the syntactic model with traditional linguistic analyses.

On the other hand, the advantage of using LLR is manyfold. Although the strength of linear co-occurrence does not necessarily correspond to the membership of syntactic constituents, a list of co-occurring tags with their statistical LLR is extremely helpful for grammar writers.

Firstly, the list focuses grammar writers' attention on really occurring phenomena. Thus, the PSG constructed in this way will not result from abstract invention of examples, but from quantifiable facts in the real corpus.

Secondly, the list provides an overview of all the distributional phenomena involved before linguists start to write the PSG. The list of all co-occurring tags can prevent linguists from committing manual omissions or errors. The relevant statistical information equips linguists with a simple and objective measure. The values of LLR highlight the strongly associated tags,

⁸ The distituent grammar in [2] contains only four rules of two tokens each. And these distituent rules do not remain accurate in every pass (or level) of construction.

providing a good set of candidates to form constituents. The values of probability (count) enable linguists to focus on phenomena which are statistically significant (i.e. with frequent occurrence).

Thirdly, when corpus are enlarged, the tag sequences and their LLR can be automatically reconstructed and compared with the old ones to show what new phenomena need to be handled in the PSG. If some modifications of rules should be made, the influences of modifications can be predicted from relevant statistical information of relevant tag sequences.

This approach, of course, may have some weak points similar to those of other corpusbased approaches. Firstly, the deduction power of the PSG will be poor with a small corpus. However, with the increasing availability of machine readable corpora, this kind of capability can be easily improved by enlarging the corpus. Moreover, if there are indeed well-known linguistic phenomena which fail to occur in the small corpus, it will be adequate for linguists to add the corresponding rules to the PSG in order to increase the descriptive power of the PSG in testing sets. Since the orginal set of constituents has been confirmed by linguists, the manual addition or modification of syntactic rules is easier to be accomplished.

Secondly, the manual category-tagging process is still too time-consuming. However, with the aid of computer tools, the tagging process can be more conveniently and systematically undertaken.⁹ Besides, once the tagged corpus is constructed, many useful models can be trained from the same corpus.

IV. Incorporating Linguistic Knowledge and Statistical LLR

In this section, our proposed cooperative approach will be illustrated by a pilot experiment of constructing a Mandarin PSG. The actual procedures are demonstrated as follows:

4.1 Constructing a Tagset

Appropriately classifying the lexical items and constructing an adequate tagset are important tasks for the whole tagging process. However, owing to the brevity of this paper, we will not pursue this issue any further, but simply present our tagset in the Appendix as a reference.

4.2 Tagging the Corpus

The sample sentences of this experiment are selected from computer technical manuals. In order to retrieve syntactical LLR from this corpus, all the sentences in this corpus have to be preprocessed. A tag will be associated to each word, representing the category (part of speech) it belongs. The LLR will be computed from the tag sequences thus obtained.

⁹ For example, the stochastic tagger proposed in [6] is an automatic tagger.

4.3 Bootstrapping

Because tagging the corpus is still a time-consuming task, we decided to start our pilot experiment with a relatively small database (2,000 sentences). In order to reduce the estimation error for sparse data, a statistical method, called "bootstrapping", is applied before LLR is computed.¹⁰ The bootstrapping method calculates the statistics over much more samples of data created by resampling from the original database. Each sample is taken independently from the original sample in order to be fair or representative of the population. In this experiment, 20,000 sentences were randomly drawn with replacement from the original 2,000 sentences to form a new database. During the sampling process, each sentence has equal chance to be selected. The new bootstrapping sample (with total number of 20,000 sentences) serves as the database for LLR calculation.

4.4 Calculating LLR from the Corpus

After applying the bootstrapping technique, the LLR of tags is automatically calculated with three different window sizes. The window size parameter allows us to look at different scales. Enlarging the window size enables linguists to build constituents with more elements. However, for the sake of reliability, the larger the window size is, the larger the corpus must be. To be compromised with the size of our database, the window sizes we chose in this experiment are 2, 3, and 4.

The formula of bigram LLR has been presented in section II. Intuitively, the original bigram LLR measure can be regarded as a measure function for a hypothesis testing problem of two events. The probability in the numerator corresponds to the event that the observed (x, y) are generated by a random source in which x and y are generated as an atom. The probability in the denominator, on the other hand, corresponds to the event that (x, y) are generated by a random source in which the generation of x and y is independent. By the same argument, the general n-gram LLR measure can also be treated as a measure function of a hypothesis testing problem. The numerator corresponds to the hypothesis that the observed data (x_1, x_2, \dots, x_n) is generated by a source in which (x_1, x_2, \dots, x_n) is generated as an atom. And, the denominator corresponds to the hypothesis that (x_1, x_2, \dots, x_n) are generated by the other sources in which the sequence x_1, x_2, \dots, x_n is generated in coincidence. The formulas with window size of 3 and 4 can thus be defined as follows:

$$LLR_{3}(x, y, z) \equiv \log_{2} \frac{P_{D}(x, y, z)}{P_{I}(x, y, z)}$$

¹⁰ Readers are referred to [10] for a review of the nonparametric estimation of statistical errors.

$$LLR_4(w, x, y, z) \equiv \log_2 \frac{P_D(w, x, y, z)}{P_I(w, x, y, z)}$$

where $P_D(x, y, z)$ is definided as the probability for x, y, z to occur jointly, and $P_I(x, y, z)$ is defined as the probability for x, y, z to occur by chance. That is:

$$P_{D}(x, y, z) \equiv P(x, y, z)$$

$$P_{I}(x, y, z) \equiv P(x) \times P(y) \times P(z)$$

$$+ P(x) \times P(y, z) + P(x, y) \times P(z)$$

Similarly, the formula for $P_D(w, x, y, z)$ and $P_I(w, x, y, z)$ are shown below:

$$P_D(w, x, y, z) \equiv P(w, x, y, z)$$

$$P_{I}(w, x, y, z) \equiv P(w) \times P(x) \times P(y) \times P(z)$$

$$+ P(w) \times P(x, y, z) + P(w, x) \times P(y, z)$$

$$+ P(w, x, y) \times P(z) + P(w) \times P(x) \times P(y, z)$$

$$+ P(w) \times P(x, y) \times P(z)$$

$$+ P(w, x) \times P(y) \times P(z)$$

We can interpret P_I as the chances that (x_1, x_2, \dots, x_n) is generated by sources which happen to be able to generate the n-gram by chance.

After computation, the number of patterns obtained with window size of 2, 3, and 4 is 451, 1893, and 4828, respectively.

4.5 Verification of the Relevance of the Groupings by Linguists

Once groups of tags have been attested with LLR, linguists will use their linguistic knowledge to decide whether these groups really form constituents or not. The information obtained in the bigram model is presented in two different forms. One is ranking the tag pairs containing the same first tag (T1) by the value of LLR, called Bigram LLR Form I. The other is ranking all the tag pairs by the value of LLR, called Bigram LLR Form II. For illustration, part of these tables are shown in Table 1 and Table 2.

T1	T2	T1_cnt	T2_cnt	T1-T2_cnt	P(T1)	P(T2)	P(T1, T2)	LLR(T1, T2)
d	cl	6299	7480	3221	0.0201	0.0239	0.01028	4.420385
d	q	6299	5844	863	0.0201	0.0187	0.00276	2.876390
d	vr	6299	270	15	0.0201	0.0009	0.00005	1.465989
d	nc	6299	72194	2110	0.0201	0.2305	0.00674	0.539350
d	a	6299	2390	12	0.0201	0.0076	0.00004	-2.001918
d	vi	6299	7171	34	0.0201	0.0229	0.00011	-2.084581
d	d	. 6299	6299	13	0.0201	0.0201	0.00004	-3.284553
d	adv	6299	16187	12	0.0201	0.0517	0.00004	-4.761671
d	vv	6299	13532	10	0.0201	0.0432	0.00003	-4.766245
d	vn	6299	9	9	0.0201	0.1131	0.00003	-6.307170

Table 1 A part of the Bigram LLR Form I (Ranking tag pairs with the same first tag by the value of LLR)

T1	T2	T1_cnt	T2_cnt	T1-T2_cnt	P(T1)	P(T2)	P(T1, T2)	LLR(T1, T2)
q	cl	5844	7480	4241	0.0187	0.0239	0.01354	4.925447
vp	р	1275	12892	1275	0.0041	0.0412	0.00407	4.602633
vxnp	р	1488	12892	1469	0.0048	0.0412	[°] 0.00469	4.584093
d	cl	6299	7480	3221	0.0201	0.0239	0.01028	4.420385
vnv	np	1073	7403	479	0.0034	0.0236	0.00153	4.239375
,	cjs	16556	10964	6239	0.0529	0.0350	0.01992	3.428368
vv	vnv	13532	1073	407	0.0432	0.0034	0.00130	3.134185
np	vv	7403	13532	2417	0.0236	0.0432	0.00772	2.917841
a	ctm	2390	21415	1210	0.0076	0.0236	0.00386	2.888484
d	q	6299	5844	863	0.0201	0.0076	0.00276	2.876390

Table 2 Top ten tag patterns in Bigram LLR Form II (Ranking all the tag pairs by the value of LLR)

Table 1 provides an overview of which tags may accompany which tags in the corpus, and equips linguists with associated statistical information. If necessary, linguists can make tradeoffs between the coverage of the grammar and the efficiency of the system by consulting the joint probabilities (or co-occurrence counts) of tag pairs. When the value of the joint probability is small, which means the real occurrences of the tag pair are few, it will be relatively adequate to ignore the distribution of the tag pair in order to reduce the complexity of the grammar and simplify the processing of the system. This table is also helpful for identifying errors in the tagged corpus or finding some important phenomena which have been overlooked by theoretical studies.

Table 2 can focus linguists' attention on strongly associated tag pairs which are more likely to be combined into constituents. To indicate tags with genuine association, patterns with LLR less than 1.0 are automatically discarded. Furthermore, because LLR becomes unreliable when the real occurrences are few, patterns with joint probabilities less than 0.0005 are also ignored.

Besides, according to linguists' intuition, certain constructions will more naturally be analyzed as tri-branching or quadri-branching instead of bi-branching. (e.g. the bi-transitive construction). Thus, the trigram model (with window size of 3) and quadrigram model (with window size of 4) will serve as convenient guides for linguists to construct constituents with more than two members. Since the list of patterns obtained in trigram and quadrigram models is too long (1893 and 4828 respectively), thresholds are also set on LLR (1.0) and joint probability (0.0005). The number of patterns thus obtained is 78 and 60 for trigram and quadrigram models respectively. These patterns are also ranked by the value of LLR. Top ten tag patterns in the trigram model and the quadrigram model are shown in Table 3 and Table 4, respectively.¹¹ It is clear that many meaningful groupings do appear in the top of these tables.¹² So, these LLR tables provide valuable clues for linguists to form constituents and help making the analyses quicker and more accurate.

T1	T2	T3	T1-T2-T3_cnt	P(T1,T2,T3)	LLR(T1,T2,T3)
(nc	}	5832	0.018619	4.582876
١	nc	Ν.	877	0.002800	4.469508
}	,	cjs	1885	0.006018	3.232748
١	nc	cjw	656	0.002094	2.950891
p	nc	vxnp	726	0.002318	2.883788
d	q	cl	744	0.002375	2.774897
р	nc	loc	2202	0.007030	2.219192
adv	vp	р	389	0.001242	2.200802
np	vv	р	724	0.002311	2.082381
р	nc	vxn	240	0.000766	2.057487

Table 3 Top ten tag patterns in the Trigram LLR Table (Ranking all the trigram tag patterns by the value of LLR)

T1	T2	T3	T4	T1-T2-T3-T4_cnt	P(T1,T2,T3,T4)	LLR(T1,T2,T3,T4)
\	nc	Ν.	cjw	293	0.000935	3.485713
р	nc	vxnp	р	726	0.002318	2.462845
np	adv	vn	ctm	1105	0.003528	2.417586
vxnp	р	nc	loc	363	0.001159	2.413632
vv	р	nc	vxnp	284	0.000907	2.397562
(nc	}	,	2086	0.006660	2.354226
q	cl	vi	ctm	412	0.001315	2.148030
}	,	cjs	vn	1353	0.004320	2.132438
vp	р	nc	loc	240	0.000766	2.055334
vn	{	nc	}	3352	0.010702	2.018944

Table 4 Top ten tag patterns in the Quadrigram LLR Table (Ranking all the Quadrigram tag patterns by the value of LLR)

¹² For example, d-q-cl is a good candidate for forming a quantifier phrase.

After having checked over the tag patterns, linguists pick out groups which should be treated as constituents, and assign phrasal tags to them. A substitution tool will automatically replace all the relevant tag patterns with new tags, or automatically locate the relevant tag patterns for linguists to confirm the substitution. Then LLR is computed again with the newly changed corpus (with new phrasal tags), yielding new LLR tables. In this experiment, we firstly constructed the quantificational phrases (Q1, consisting of "(d) (q) (cl)"), the low level coordinate phrases, and substituted "{ nc }" with N0. Part of the resulting new tag patterns in different n-gram models are shown in Table 5, Table 6, and Table 7.

T 1	T2	T1_cnt	T2_cnt	T1-T2_cnt	P(T1)	P(T2)	P(T1, T2)	LLR(T1, T2)
q	cl	281	417	271	0.0010	0.0015	0.00098	9.323793
d	cl	296	417	146	0.0011	0.0015	0.00053	8.356441
ŃJ	ctn	3894	205	141	0.0141	0.0007	0.00051	5.613008
vp	р	1275	12892	1275	0.0046	0.0465	0.00460	4.425786
vxnp	р	1488	12892	1469	0.0054	0.0465	0.00530	4.407246
vnv	np	1073	7403	479	0.0039	0.0267	0.00173	4.062527
vnp	р	259	12892	163	0.0009	0.0465	0.00059	3.757706
adv	vns	16209	222	175	0.0585	0.0008	0.00063	3.752262
,	cjs	16485	10945	6233	0.0595	0.0395	0.02249	3.258835
VN0	Q1	496	10467	148	0.0018	0.0378	0.00053	2.981671

Table 5 LEVEL II Top ten tag patterns in Bigram LLR Form II (Ranking all the tag pairs by the value of LLR)

T 1	T2	T3	T1-T2-T3_cnt	P(T1,T2,T3)	LLR(T1,T2,T3)
cjs	vv	VNJ	342	0.001234	3.253585
NO	,	cjs	1922	0.006936	3.062737
vp	р	N0	229	0.000826	2.838479
р	nc	vxnp	726	0.002620	2.829509
р	N0	loc	419	0.001512	2.738720
vv	vnv	np	230	0.000830	2.688979
	adv	vp	206	0.000743	2.271391
р	nc	loc	2202	0.007947	2.178622
р	nc	vxn	240	0.000866	2.095507
vn	NO	,	2086	0.007528	2.085268

Table 6 LEVEL II Top ten tag patterns in the Trigram LLR Table (Ranking all the trigram tag patterns by the value of LLR)

Linguists then check the new tag patterns to look for higher level constituents. This procedure is recursively applied until there is only one phrasal tag (S) left in every sentence. A complete PSG for this corpus is thus obtained.

When the size of the corpus is small, many constructions may not be included in this corpus. Thus, they will fail to appear in the LLR tables. However, if linguists are aware of their importance in the applicational domain, and there are indeed well-justified linguistic analyses

T 1	T2	T3	T4	T1-T2-T3-T4_cnt	P(T1,T2,T3,T4)	LLR(T1,T2,T3,T4)
NO	,	cjs	VNJ	193	0.000697	3.424608
adv	vp	р	NO	149	0.000538	2.573180
р	Q1	nc	vxnp	141	0.000509	2.434070
р	nc	vxnp	р	726	0.002620	2.374187
vxnp	р	nc	loc	363	0.001310	2.335425
VNJ	nc	,	adv	232	0.000837	2.284488
. vv	р	nc	vxnp	284	0.001025	2.257929
np	adv	vn	ctm	1105	0.003988	2.233253
vn	N0	,	cjs	1885	0.006803	2.184932
vp	р	nc	loc	240	0.000866	1.975087

Table 7 LEVEL II Top ten tag patterns in the Quadrigram LLR Table(Ranking all the Quadrigram tag patterns by the value of LLR)

for them, it will be convenient for linguists to directly incorporate the existing analyses into the PSG. The descriptive power of the PSG for testing sets can be enlarged by incorporating linguistic knowledge in this way.

V. Conclusion

This paper discusses several methods of constructing a PSG, including the purely linguistic approach, the purely automatic approach, and the proposed human-machine cooperative approach. The advantages of the proposed approach over other methods are briefly summarized as follows:

- 1. The corpus-based statistical approach focuses linguists' attention on authentic material instead of invented examples.
- 2. The LLR tables equip linguists with an overview of the distributional phenomena involved, preventing linguists from committing manual errors or omissions.
- 3. The LLR statistic highlights the strongly associated tag sequences, providing a good set of candidates for forming constituents.
- The statistical information provides an objective measure for linguists to make tradeoffs, and enables linguists to focus on phenomena of statistical importance rather than of theoretical interest.
- 5. When modifications are made, the tagged corpus and the relevant statistical information can be automatically and systematically reconstructed.
- 6. The syntactic model can be coupled with traditional semantic models.

7. The grammar is able to incorporate achievements of linguistic researches.

According to our experience in the pilot experiment, the LLR statistic really helps making the analyses quicker and more accurate. As most of us believe, human grammar writers could do a better job if they had access to better tools. LLR statistic is suggested in this paper as the right tool.

REFERENCES

- [1]. Anick, P., and J. Pustejovsky, "An Application of Lexical Semantics to Knowledge Acquisition from Corpora," *Coling 90*, vol. 2, 7-12, 1990.
- [2] Brill, Eric, David Magerman, Mitchell Marcus, and Beatrice Santorini, "Deducing Linguistic Structure from the Statistics of Large Corpora," *Proceedings of the Third DARPA Workshop on Speech and Natural Language*, U.S.A, 1990.
- [3] Brown, Peter F., John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin, "A Statistical Approach to Machine Translation," *Computational Linguistics*, vol. 16, 79-85, 1990.
- [4] Chao, Yuen-Ren, A Grammar of Spoken Chinese, Berkeley: University of California Press, 1968.
- [5] Chinese Knowledge Information Processing (CKIP), Kuo2 Yu3 De0 Tsi2 Lei4 Fen1 Shi1 (Shioul Ding4 Ban3), Nankang: Academia Sinica, 1988.
- [6] Church, Kenneth, "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text," Proceedings of Second Conference on Applied Natural Language Processing, Austin, Texas, 1988.
- [7] Church, Kenneth and Patrick Hanks, "Word Association Norms, Mutual Information, and Lexicography," *Proceedings of ACL-27*, Vancouver, 76-83, 1989.
- [8] Church, Kenneth, William Gale, Patrick Hanks, and Donald Hindle, "Parsing, Word Associations, and Typical Predicate-Argument Relations," *Proceedings of the International Workshop* on Parsing Technologies, C.M.U, 1989.
- [9] Church, Kenneth, William Gale, Patrick Hanks, and Donald Hindle, "Using Statistics in Lexical Analysis," in U.Zernik (ed.), *Lexical Acquisition: Exploiting On-Line Resources*, Hillsdale, N.J. Lawrence: Erlbaum Associates, 1990.
- [10] Efron, Bradley and Gail Gong, "A Leisurely Look at the Bootstrap, the Jackknife, and Cross-Validation," *The American Statistician*, vol 37, No. 1, 36-48, 1983.
- [11] Fu, King-Sun, "Syntactic Pattern Recognition and Applications," Englewood Cliffs, N.J.: Prentice-Hall, Inc, 1982.
- [12] Garside, Roger, Geoffrey Leech and Geoffrey Sampson, The Computational Analysis of English : A Corpus-Based Approach, London: Longman, 1987.

- [13] Hopcroft, John E. and Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, U.S.A.: Addison-Wesley, 1979.
- [14] Hsu, Yu-Ling and Keh-Yih Su, "Criteria for the Classification of Lexical Categories in a Syntax-Oriented Parsing System," *Proceedings of ROCLING I*, 215-227, 1988.
- [15] Hsu, Yu-Ling and Keh-Yih Su, "Lexical Categorization in a Syntax-Oriented Parsing System," In preparation.
- [16] Su, Keh-Yih and Jing-Shin Chang, "Some Key Issues in Designing MT Systems," *Machine Translation 5*, 265-300, 1990.
- [17] Tou, Julius T. and Rafael C. Gonzalez, Pattern Recognition Principles, U.S.A.: Addison-Wesley, 1974.
- [18] Zernik, U. and P. Jacobs, "Tagging for Learning : Collecting Thematic Relations from Corpus", *Coling 90*, vol 1, 34-39, 1990.

APPENDIX

The tagset used in this experiment is listed below (punctuation marks are not included). Readers are referred to [4], [5], [14], and [15] for detailed discussions on lexical categorization.

- nc : common nouns
- np : proper names or pronouns
- d : determiners
- q : quantifiers
- cl : classifiers
- p : prepositions
- loc : locatives
- ref : reflexives

vi : intransitive verbs

- vn : verbs followed by a single nominal argument
- vnn : verbs followed by double nominal arguments
- vs : verbs followed by a sentential argument
- vv : verbs followed by a verbal argument
- vp : verbs followed by a prepositional phrase argument
- vr : verbs introducing an obligatory relative clause
- vns : verbs followed by nominal and sentential arguments
- vnv : verbs followed by nominal and verbal arguments
- vnp : verbs followed by a nominal object and a prepositional clause argument
- vxn : verbs preceded by a preposed nominal object
- vxnn : verbs preceded by a preposed nominal object and followed by a second object
- vxnp : verbs preceded by a preposed nominal object and followed by a prepositional phrase argument
- vxnv : verbs preceded by a preposed nominal object and followed by a verbal object
- vxns : verbs preceded by a preposed nominal object and followed by a sentential object
- a : adjectives
- asp : aspect markers
- adv : adverbs
- cjs : conjunctions for sentences
- cjv : conjunctions for verb phrases

274

cjw : conjunctions for words or other phrases

ctm : modifier clitics

cts : sentential clitics

ctn : noun clitics

excl : exclamatives
