パラメータによる日本語連体修飾構造の解析

Timothy Baldwin, 徳永 健伸, 田中 穂積

東京工業大学 情報理工学研究科

〒 152-8552 東京都目黒区大岡山 2-12-1

{tim,take,tanaka}@cl.cs.titech.ac.jp

概要

本稿では，従来の手続き的な日本語連体修飾解析法と，同一パラメータを用いたC4.5による解析を比較し，従来の解析法の実用性を評価する．C4.5用にデータを加工するときに，節内解釈の曖昧性を解消する手法と，節間の解釈を統一させる手法をいろいろと提案し，それぞれを評価する．さらに，従来の解析法に使用したパラメータの組合せの定量的な評価を行ない，用言意味属性によるパラメータスペースの拡張を試みる．最終評価では，C4.5による解析で89%の精度が得られ，従来の解析法を若干上回った．C4.5によって推測されたルールセットと従来の解析法に使われたルールセットの構成や頑健性を比べたところ，著しく類似していることがわかった．

The parameter-based analysis of Japanese relative clause constructions

Timothy Baldwin, Takenobu Tokunaga and Hozumi Tanaka

Dep. of Computer Science, Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552

{tim,take,tanaka}@cl.cs.titech.ac.jp

Abstract

We examine the validity of a procedural Japanese relative clause analysis system by way of running C4.5 over the same basic parameter space and comparing results. In reformatting data for use with C4.5, we propose and test various ways of reducing intra-clausal interpretational ambiguity and cross indexing the overall analysis for the relative clause construction across coordinated relative clauses. We additionally investigate the disambiguating effect of the different parameters utilised in the original system, and go on to complement the parameter space with verb semantic attributes. In final evaluation, C4.5 marginally outperforms the procedural system formulation, returning an accuracy of around 89% for the most successful system configuration. Comparison of the robustness and rule composition of the original system and optimal rule set proposed by C4.5 revealed striking similarities.

1 Introduction

Given a taxonomy of Japanese relative clause construction types and a basic corpus of Japanese relative clause construction instances, we investigate the success of various parameter configurations in classifying relative clause constructions. The system of relative clause construction ("RCC") types was originally devised in Baldwin (1998), in addition to proposing a set of lexical and semantic parameters to characterise RCC's according to the proposed typology; the proposed parameter set was implemented in the form of a procedural rule set to determine RCC type.

Validation of the original system is achieved by running the C4.5 decision tree-based classification system (Quinlan, 1993) over the same set of features as was utilised in the original research. In this, we seek to validate both the rule ordering and feature composition. At the same time, we attempt to ascertain a ceiling on system performance for the given set of parameters, and identify shortfalls in the given parameter description. We further go on to look at the potential for augmentation of the original parameterisation with verb semantic attributes (Nakaiwa et al., 1994; Nakaiwa and Ikehara, 1997).

As with many tasks relating to natural language, the parametric characterisation of RCC's is dogged by analytical ambiguity, in particular for word sense, phrase boundary and phrase head ambiguity. The latter two of these concerns are resolved by pre-processing data into phrase units according to the original corpus mark-up (the EDR corpus (EDR, 1995), in our case), leaving the question of word sense ambiguity. Given that we are keen to minimise the cost of the rule formulation, we largely avoid the need for verb sense disambiguation by associating a unique case frame with each verb stem type. Even here, however, we must have some means of dealing with verb homonymy and complex relative clauses. We investigate various techniques to resolve such ambiguity and combine the analysis of multiple component clauses.

One feature of the original system is that it is designed for shallow, low-cost analysis, centring principally around a basic case frame and verb class description. That is, we avoid consideration of case slot-specific selectional restrictions and pragmatics—as are suggested to heavily influence RCC construal—in proposing a fast, lightweight analysis method. Such processing is suggested to have applications for machine translation from Japanese, in determining the semantic type of the RCC for transferral across to the target language. It also has a place in any information extraction or text understanding task in determining the semantic relation between the head noun and modifying relative clause.

In the proceeding sections, we first define the nature of Japanese RCC's (Section 2) and outline the architecture of the original analysis system (Section 3). Next, we describe how the original system translates across to a C4.5-based implementation (Section 4), before evaluating various system configurations and disambiguation techniques (Section 5). We conclude with a discussion of the ramifications of the presented results (Section 6).

2 Definitions

Japanese **relative clause constructions** (RCC's) are defined as being NP's of structure `[[S][NP]]`, noting the lack of a relative pronoun or any other explicit form of noun–clause demarkation. Japanese relative clauses have finite inflection and are in all respects syntactically identical to matrix clauses. Relative clause modification occurs in three major semantic categories, indistinguishable lexically: case-slot gapping, head restrictive and idiomatic. With **case-slot gapping** RCC's (aka 'inner' relative clauses (Teramura, 1975 78) or 'clause host' constructions (Matsumoto, 1997)), the head NP can be considered to have been gapped from a case slot subcategorised by the main verb of the relative clause. Note here that, whereas the case slot from which gapping has occurred tends to have a distinctive case marking schema, that marking is not preserved either within the relative clause or on the head NP. **Head restrictive** RCC's (aka 'outer' relative clauses (Teramura, 1975 78) or 'noun host' constructions (Matsumoto, 1997)) occur when the relative clause modifies or restricts the denotatum of the head NP. **Idiomatic** RCC's are produced when the overall relative clause construction produces an idiomatic reading. Examples of the three RCC types are, respectively:[1]

(1) *kinō* *katta* *bōsi*
yesterday bought hat
"the hat ( ) bought yesterday"

(2) *bōsi-o* *katta* *riyū*
hat-ACC bought reason
"the reason ( ) bought a hat"

(3) *hito-o* *miru* *me*
person-ACC see eye
"the ability to judge a person"

The inherent difficulty in determining the type of RCC construal comes from the fact that these 3 categories of RCC construal and the 26 RCC sub-types contained by them are syntactically identical. We thus have no option but to consider each construal type on its individual merits for every RCC input.

For our purposes, case-role gapping is considered to occur in nineteen sub-categories, such as: SUBJECT, DIRECT OBJECT, PASSIVE AGENT, CO-ACTOR, LOCAL ABLATIVE, LOCATIVE and TEMPORAL; this inventory coincides with the case-role markers used for case slots. Note that RCC (1) above is a DIRECT OBJECT case-role gapping RCC. In our case-role set, syntactic markers such as subject, indirect object and passive agent override the more conventional case-role descriptors of agent and patient, in the case that a given case slot is subject to grammatical processes. Our motivation in this is that the case-role gap in the case of coordinated case-role gapping relative clauses is governed along syntactic rather than case-role semantic lines; additionally, the use of grammatical relations allows us to model the type of accessibility hierarchy as described in Keenan and Comrie (1977)/Silverstein (1976) and Inoue (1976), whereby items higher up in the hierarchy are more readily gapped.

Case-role gapping can also be realised by way of binding or possession of an instantiated case slot, over the full range of case slot types; we term such RCC modification as **binding**. In the current formulation, due to the infrequency of BOUND RCC's (just over 1% of all RCC's observed in evaluation), we simply identify BOUND RCC's as such, without description of the actual case slot which the head noun binds. An example of a BOUND RCC (with binding on the subject position) is:

(4) *pēzi-ga* *otite-iru* *hon*
pages-NOM are missing book
"a book with missing pages"

Head restrictive RCC's come in six varieties according to the nature of modification, namely: DEGREE, EXCLUSIVE, INCLUSIVE, GENERAL RESTRICTIVE, RELATIVE TEMPORAL and RESULTATIVE. For details, the reader is referred to Baldwin (1998). By way of note, we classify (2) above as being GENERAL RESTRICTIVE.

IDIOM RCC's are treated as forming a single class.

3 The original formulation

The original system described in Baldwin (1998) is powered by a hand-crafted rule set, designed with the intent to

[1] The following nomenclature is employed in example sentences throughout this paper: NOM = nominative, ACC = accusative, PRES = non-past, ( ) = zero argument

evaluate the efficacy of shallow processing on RCC analysis. Relative clause construal has traditionally been portrayed as a largely semantic and pragmatic affair, a claim which we set out to dispute empirically by producing an essentially lexical system with high accuracy.

3.1 Parameter description

Parameters employed in the system include: a generalised case frame description, a verb class characterisation, verb inflectional analysis, basic noun semantics and various trigger patterns. These are encoded in the form of a procedural rule set, producing a single output for each activated dictionary entry.

Case frames are applied in determining which core case slots are instantiated and hence *un*available for case-role gapping, and conversely which case slots are *un*instantiated and available for case-role gapping. Fixed expressions are governed by the constraint that all fixed case slots must be instantiated in the input for that case frame to be triggered, including the possibility of fixed arguments being expressed as the head noun of the RCC.[2]

Case frames were generated from the Goi-Taikei pattern-based valency dictionary (Ikehara et al., 1997) by conflating the major senses for each distinct verb stem (distinct kanji–reading pairing). In essence, case frames are simply a list of the 'core' case slots for the verb in question in their canonical ordering, with each case slot being marked for canonical case marking and case-role (with case-roles taking the form described above for case-role gapping—see Section 2). In the case of case frame-transforming verbal inflection, the basic case frame is manipulated by way of automated rules to produce a final surface case frame for use in processing.

The minimalistic case frame description is complemented by **verb classes**. Verb classes are used to describe such effects as adjunct compatibility (no adjunct case slots are contained within case frames), case slot interaction, potential for valency-modifying alternation, and compatibility with particular lexical trigger patterns. Due to the generally orthogonal nature of the verb classes, each verb/case frame entry generally receives multiple verb classes. Examples of verb classes are *excluding* verbs, associated with a distinctive trigger pattern producing EXCLUSIVE RCC construal, and *action* verbs, compatible with a locative case slot.

The **inflectional analysis** of a verb produces an ordered list of inflectional features, including tense, aspect and voice. These have applications in case frame transformation, as trigger conditions for various analysis types, and in the scoring of individual clause interpretations.

Basic noun semantics are used to (a) semantically classify the head noun of the RCC, and (b) filter out locative and temporal case slots from the relative clause. Head nouns are classified according to the binary vector of ±`agentive`, ±`1st_person_pronoun`, ±`pronoun`, ±`instrumental`, ±`local`, ±`temporal`, ±`durational`, ±`abstract`, ±`non_gapping` and ±`degree`. As we have no means of disambiguating noun sense, this characterisation corresponds to the union of features of all senses of the head noun, as defined within the Goi-Taikei thesaurus (Ikehara et al., 1997). That locative and temporal case slots should require filtering off is a direct consequence of our shallow processing method, and simple case marker matching mechanism to determine case slot instantiation. Non-adjunct and adjunct case slots can overlap in case marker mark-up, and by filtering off adjunct case slots, we reduce the scope for error in this matching process.

[2]Only a small proportion of fixed arguments can, in fact, be gapped to the head noun position, with potential for gapping being determined by factors such as the semantic transparency of the fixed argument. We model this variability by way of 'displaceability' judgements on each fixed argument, within the case frame dictionary.

An example of a **trigger pattern**, in the case of *excluding* verbs, is the combination of simple past or non-past main verb inflection, and the occurrence of only an accusative-marked case slot within the relative clause. The satisfaction of these constraints produces the EXCLUSIVE analysis type:

```
IF (excluding-type verb AND simple main
verb inflection AND unique accusatively
marked argument) RETURN EXCLUSIVE
```

The EXCLUSIVE analysis type is thus produced for RCC (5) below.

(5) *nitiyōbi-o* *nozo-ku* *mainiti*
Sunday-ACC exclude-PRES everyday
"everyday except Sundays"

3.2 Analytical multiplicity

Multiple clause analyses arise in the case of verb homophony/homography and for fixed expressions (*intra-clausal ambiguity*), as well as for relative clause coordination (*inter-clausal cross-indexing*).

For the purposes of our system, **verb homophony** refers to the state of multiple verb entries in the case frame dictionary sharing the same kana content (and hence pronunciation), whereas **verb homography** occurs when multiple verb entries coincide in kanji content. An example of verb homophony is seen for the verbs 合う "to correspond" and 会う "to meet", both pronounced as *au*, while an example of verb homography is seen for the verbs *tomeru* "to stop" and *yameru* "to quit", both expressed as 止める. The partial overlap in lexical form leads to the situation of multiple verb entries being triggered, producing independent analyses for the RCC input.

Fixed expressions include such constructs as *ma-ni au* "to make it on time". They produce multiple analyses due to the verb employed in the fixed expression tending also to have a generalised usage, as occurs for the *au* in our example of *ma-ni au*. Hence, the fixed expression and generalised usages produce separate RCC construal analyses, and we require some mechanism to choose between them. This is achieved in the first by preferring analyses stemming from fixed expressions, over those deriving from verb class-based trigger patterns, in turn over those generated through generalised techniques. We define each such stratum as comprising a distinct **expressional type**.

In the case that ambiguity is not resolved through such *a priori* expressional type preferences, we score each clause interpretation by way of the **representational preference** for the current verb to take different lexical forms. The representational preference (RP) of lexical form a of verb entry f (i.e. a_f) is defined as the likelihood of f being realised as a, with a median score of 1.

$$RP(a_f) = \frac{1+freq(a_f)}{1+\sum_{i \neq a} freq(i_f)} \quad (1)$$

This is normalised over the representational preference for all source entries a_i, to produce the **normalised representational preference** $NRP(a_f)$.

$$NRP(a_f) = \frac{RP(a_f)}{\sum_i RP(a_i)} \quad (2)$$

We further introduce the notion of **complexity of inflectional content** (CIC) to add in penalisation of inflectionally complex analyses. CIC is computed *in situ* based on the number of inflectional morphemes contained in the verb, relative to the parse of simplest inflectional content (min_infl); the simplest parse receives a complexity of one. CIC is combined with NRP to produce an overall verb score VS for lexicalisation a of verb f:

$$VS(a_f) = \frac{NRP(a_f)}{(CIC(a_f) - min\_infl + 1)^{\alpha}} \quad (3)$$

Here, α is a constant weighting factor, used to adjust the degree of penalisation of inflectional complexity. In evaluation, α was set to 2.

The various VS scores for entries producing a common analysis are added together, and the analysis with the highest combined score selected as the unique system output; in the case of a tie, we randomly pick one of the highest-ranking analyses. Note that 28.8% of clauses occurring in the evaluation data are associated with analytical ambiguity.

The above methods are applicable to **intra-clausal disambiguation**, which is performed prior to **inter-clausal cross-indexing**. For the case of the coordinated RCC `[[S1 S2] NP]`, therefore, we individually disambiguate `S1` and `S2`, and apply the final clausal interpretations in inter-clausal cross-indexing between `S1` and `S2`. Inter-clausal cross-indexing is relevant in cases of relative clause coordination.

Relative clause coordination occurs when the relative clause is composed of two or more coordinated unit clauses, as was observed for 4.7% of the RCC's targeted in evaluation. While it would certainly be possible to ignore all other than the final clause and allow this to determine our overall analysis (as we test in evaluation below), by considering all component clauses, we are able to apply the constraint that component clauses tend to agree in analysis type.[3] That is, it does not happen that we have one SUBJECT case-role gapping clause and one GENERAL RESTRICTIVE clause, for example, coordinated within a single RCC. For case-role gapping RCC's, agreement of analysis type occurs according to grammatical relations, such that we can have coordinated active and passive clauses producing an overall SUBJECT case-role gapping RCC; this is one reason for our choice of grammatical relations as the descriptor for case slots.

A single overall analysis type is determined by way of combining all individual clausal interpretations of maximal expressional type, as for verb homophony/homography. For inter-clausal cross-indexing, however, we additionally block certain analysis types explicitly disallowed in any of the component clauses. This is achieved simply by deleting the disallowed analysis types from the final list of scored outputs.

3.3 Limitations

When run over the 5143 RCC instances targeted in evaluation, the original RCC analysis system performed at a creditable accuracy of around 87.6% (cf. the 88.6% accuracy quoted in Baldwin (1998) over a slightly smaller data set). However, several issues regarding its implementation and extendibility remain unanswered.

First and foremost is the optimality of the proposed rule set over the given parameter set. That is, would it be possible to produce better performance for different rule orderings or parameter combinations? Related to this is the question of the rule set implementation over-fitting the data on which the system was evaluated: would the system perform comparably on truly unseen data?

In answering these questions, we look to determine a performance ceiling for the given parameter set (and hence the type of surface analysis we are targeting), investigate the possibilities of expanding the parameter set, and examine different methods of resolving ambiguity within the given domain.

This validation of the original formulation is performed by way of the C4.5 decision tree-based classification system (Quinlan, 1993).

[3]The principal exception to this constraint observed in data is for coordinated BOUND and strict case-role gapping relative clauses. Occurrences of such RCC's are infrequent enough, however, to be able to apply our constraint with high reliability.

4 C4.5-based implementation

C4.5 (Quinlan, 1993) is a decision tree-based classification system which has seen prominent applications within natural language processing in automatic verbal case frame acquisition (Almuallim et al., 1994; Tanaka, 1996) and ellipsis resolution (Yamamoto and Sumita, 1998). Essentially, C4.5 takes a set of feature vectors of pre-determined format as input, and induces a decision tree which characterises the given parameter space.

4.1 Parameter set

So as to be able to run C4.5 over the same parameter set as for the original system, we clearly need to identify the exact set of parameters and nature of diagnostics employed in the original system. Additionally, so as to make system comparison fair, we want to encode parameters not simply as the individual lexical and semantic conditions relied upon in trigger patterns, but as the compatibility of the input RCC with those trigger patterns. That is, we want to evaluate not C4.5's ability to learn the carefully devised set of trigger patterns, but the optimal ordering of those trigger patterns. At the same time, many individual conditions relied upon in trigger patterns are retained in the parameterisation. One point of interest is whether C4.5 will be able to postulate any novel parameter clusters as rules of wide applicability.

Both individual features and trigger pattern compatibility judgements are described in the main as binary values. As for the original system, individual features are basically case slot compatibility, head noun semantic and verb class membership flags. Case slot compatibility flags indicate that the case slot in question is both contained within the case frame for the main verb of that clause, and non-instantiated. Only those (non-adjunct) case slots finding their way into case frames are described in this way, with adjunct case slots described implicitly by way of verb class membership. Examples of trigger patterns described through compatibility flags are those for the *excluding* verb class and RELATIVE TEMPORAL construal type. A value of 1 designates the trigger pattern as having being satisfied. Note that both verb classes and head noun semantic features form a partial hierarchy, such that the activation of certain features will automatically produce the activation of other ancestor features, as occurs for `+1st_person_pronoun` acting on `+pronoun`.

The only parameter not described via a binary value is that for gapped fixed arguments, which takes a value of zero in the case that the head noun does not constitute a gapped fixed argument, or the case-role of the gapped fixed argument if there is the possibility of a gapped fixed argument analysis.

4.2 Data extraction

Evaluation of the RCC corpus utilised in development of the original system was achieved through conversion of each RCC into a feature vector corresponding to the parameter set described above. While this is a relatively trivial process for simple RCC's where the main verb of the relative clause has a unique interpretation,[4] coordinated RCC's and relative clauses with multiple intra-clausal interpretations of equivalent expressional strength pose a more subtle problem. Here, we can either attempt to partially resolve or preserve ambiguity in a single feature vector and apply C4.5 conventionally, or run C4.5 on the individual feature vectors described by each interpretation and use error estimation to select the most plausible analysis type. We opt for the first of these alternatives, and look at various means of selecting the single most plausible interpretation or combining features of the various

[4]This assumes the same first-line defence against expressional type ambiguity as was described for the original system, whereby fixed expressions are given preference over trigger patterns, which in turn override general analyses.

candidates (*intra-clausal parameterisation*); we also separately consider retaining individual feature vectors for coordinated clauses or integrating them into one overall descriptor for the entire RCC (*inter-clausal parameterisation*). As for the original system, we choose to process intra-clausal ambiguity first, and then apply the resultant analysis in inter-clausal cross-indexing.

The primary method tested for selecting the single 'best' clause interpretation at a given expressional level is VS as described above. An alternative method tentatively trialed to remove interpretational ambiguity is to collapse the feature vectors deriving from the various interpretations into a single feature vector, through logically `OR`'ing the vectors together.[5] In this way, we are effectively retaining all possible interpretations and having C4.5 select the salient features from among them.

In the instance of a coordinated relative clause, we again have a choice as to whether to combine the individual clause interpretations into a single **clause-integrated** feature vector and constrain the scope of interpretation appropriately, or maintain individual **unit clause** feature vectors for the component relative clauses, cross-indexing them in some way; each unit clause feature vector is given the analysis type for the overall RCC. A clause-integrated feature vector can be attained simply by `AND`'ing or `OR`'ing the component clause interpretations together. `AND`'ing has the advantage of forcibly constraining the overall interpretation through disallowing all case-role gapping interpretations where that case-role is either instantiated in one of the component clauses or not contained within a case frame. `OR`'ing, on the other hand, avoids potentially erroneous over-constraint of interpretation, and places the onus of analytical discrimination on C4.5.

A potential source of disambiguation largely unutilised in the original system, is the verb semantic attribute ("VSA") annotation from the Goi-Taikei pattern-based valency dictionary, which describes the basic semantics of the verb (Nakaiwa et al., 1994; Nakaiwa and Ikehara, 1997). Examples of VSA's are *perceptual state* and *emotive action*. For the purposes of this paper, VSA's can be considered to be orthogonal to our verb classes, as they target the type of activity or state described by the verb, whereas our verb classes relate to case slot interaction and compatibility with highly specialised trigger patterns. VSA's were retained in the case frame dictionary simply by taking the union of all VSA's for those verb entries used to form the case frame dictionary entry. This dilutes the discriminatory power of VSA's somewhat (from around 1.07 to over 1.35 VSA's per non-fixed sense dictionary entry, out of a total of 36 attribute types), but not so as to make them completely homogeneous. In evaluation, we provisionally test the applicability of VSA's in an attempt to further enhance the original system, with the caveat that such application does not reflect the true potential of VSA's.

5 Evaluation

In evaluation, we variously compare: (a) combinations of the intra-clausal interpretation selection techniques described above; (b) clause-integrated vs. unit clause feature vectors for coordinated relative clauses; and (c) the applicability of VSA's to the resultant system configurations. We further go on to investigate the efficacy of different parameter partitions on disambiguation, and generate a learning curve of system performance over data sets of increasing size. We additionally perform straight evaluation of the original system, and conclude by testing the comparative performances of the original and C4.5-based systems on a set of 100 entirely new RCC instances.

Without exception, evaluation with C4.5 was carried out by way of 10-fold cross validation, with the pruning constraint set to 10%.

The data used in evaluation is a set of 5143 RCC instances extracted from the EDR corpus (EDR, 1995); these 5143 RCC's comprise a total of 5408 matrix relative clauses (through coordination). We thus have 5143 inputs for the original system, 5143 clause-integrated feature vectors and 5408 unit clause feature vectors.

As with any evaluation, we need a baseline with which to benchmark the performance data. An absolute benchmark on accuracy is obtained through allotting a SUBJECT case-role gapping analysis to every RCC input, based on the absolute and relative frequencies of the ten most common analysis types in the 5143 clause-integrated feature vectors as given below. We attain a baseline accuracy of 64.7% in this way.

Class	***Freq.***	***Rel. freq.***
SUBJECT gapping	3328	0.647
GENERAL RESTRICTIVE	653	0.127
DIR. OBJECT gapping	373	0.073
IDIOM	131	0.026
EXCLUSIVE	117	0.023
LOCATIVE gapping	113	0.022
TEMPORAL gapping	105	0.020
CO-ACTOR gapping	63	0.012
BOUND gapping	55	0.011
TIME DURATIVE gapping	51	0.010

5.1 Intra-clausal disambiguation

Intra-clausal disambiguation refers to the selection of the most plausible interpretation for the given clause, in the case of ambiguity. Here, we compare: (a) a random selection baseline method (*UC+best_rand*); (b) a method where all feature vectors for the current clause are logically OR'ed together (*UC+or*); and (c) the VS maximisation method proposed in the original research (*UC+vs*). The results for the various methods within a basic unit clause feature vector framework (with no interaction between clause analyses) are presented to the left of Figure 1. Note that the presented figures are for the entire data set, despite intra-clausal analytical ambiguity arising for only 28.8% of unit relative clauses.

UC+vs outperforms the *UC+best_rand* baseline to a level of statistical significance, in both training and testing. *UC+or* outperforms the baseline *UC+best_rand* method, but lags behind *UC+vs* in testing in particular, but also training and only holds a statistically significant edge over *UC+best_rand* in training. The relatively strong performance for *UC+or* in training suggests that the given data size is perhaps insufficient for it to perform to its true potential, but it is still slightly down on the performance of *UC+vs*.

Based on the above results, we choose *UC+vs* as our intra-clausal disambiguation technique for all subsequent evaluation.

5.2 Inter-clausal cross-indexing

Next, we look at the different inter-clausal analysis methods. The two core paradigms we consider are unit clause (*UC*) and clause-integrated (*CI*) analysis.

For unit clause analysis, we test two methods, the first being simple unit clause disambiguation in the form of *UC+vs* from above, and the second being an extension of this basic methodology, in which we constrain the scope of case-role gapping by logically `AND`'ing together the case slot compatibility flags between unit clause feature vectors (*UC+vs**); note that VS is applied as is for intra-clausal disambiguation in the second case.

[5] Note that for the gapped fixed argument feature—the only non-binary feature described—no instances of multiple non-zero values were observed in evaluation. We thus only have to consider the conventional `OR`'ing of two zeros together, and the `OR`'ing of a zero and non-zero gapped fixed argument value. In the latter case, we simply return the non-zero (case-role) value.

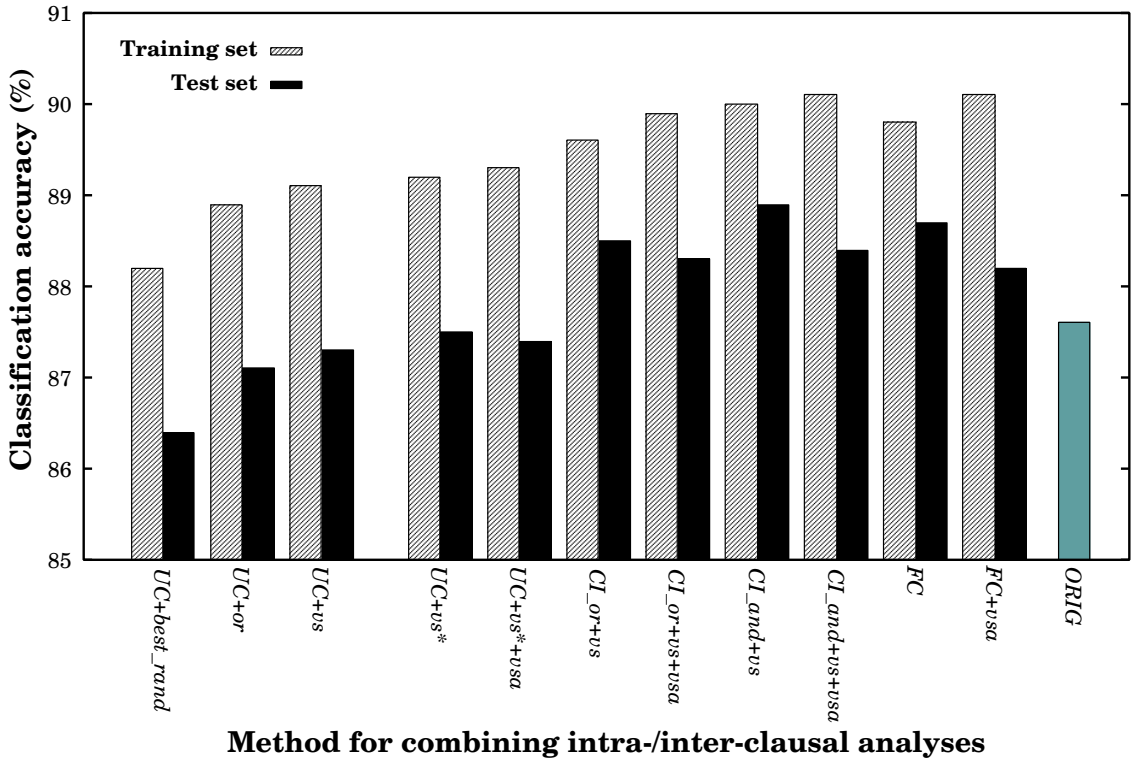

Figure 1: Evaluation of the different clause processing configurations

For clause-integrated analysis, we again apply VS in intra-clausal analysis, then either logically `OR` or `AND` the component unit clause feature vectors together, producing methods *CI_or+vs* and *CI_and+vs*, respectively.

To benchmark these integrated approaches, we test the accuracy of simple final clause analysis for coordinated relative clauses (FC). Here, we simply ignore all other than the final clause in both the training and testing phases, and perform intra-clause disambiguation with VS as for the other methods.

We go on to produce an additional variant of methods *UC+vs**, *CI_or+vs*, *CI_and+vs* and *FC* by adding in to the feature vector the VSA content for each selected unit clause interpretation, leading to *UC+vs*+vsa*, *CI_or+vs+vsa*, *CI_and+vs+vsa* and *FC+vsa*, respectively. Note that VSA's are also the target of logical operations for *CI_or+vs+vsa* and *CI_and+vs+vsa*.

The training and test accuracies for the described methods are given in Figure 1, juxtaposed against the 87.6% accuracy attained for the original system (labelled as *ORIG*). As for intra-clausal disambiguation, the presented evaluation is over the entire data set, despite relative clause coordination occurring for only 4.7% of inputs. It is not expected, therefore, that the addition of inter-clausal disambiguation will hugely affect performance.

In both training and testing, the two clause-integrated analysis methods outstrip the unit clause analysis methods to varying degrees, with the superior method being *CI_and+vs* at a test accuracy of 88.9%. In training, all visible disparities in accuracy between VSA and non-VSA methods other than that between *CI_and+vs+vsa* and *FC+vsa*, are statistically significant according to the t-test ($\alpha \leq 0.05$). For test accuracies, on the other hand, a statistically significant performance improvement was seen only for *CI_and+vs±vsa* over *UC+vs*±vsa* and *FC* over *UC+vs**. As such, the absolute superiority of *CI_and+vs* is somewhat doubtful for the given data size, but it can be expected to produce genuine performance gains given greater data. These figures are particularly promising given the relative scarceness of coordinated RCC's in the input data.

The slight disparity in training accuracies of system configurations with VSA's over those without,[6] and drop in test accuracy over non-VSA data sets, would tend to suggest that the given data set is too small to bring out the full disambiguating power of VSA's, and that they have potential to tweak the system performance marginally, assuming sufficient data. Without VSA's, even, the training accuracy of 90.0% for *CI_and+vs* can be perceived as a ceiling on optimal system performance for the presented parameter formulation without VSA's.

It is difficult to gauge the significance of the results given that coordinating RCC's account for only 4.7% of the total data. One way in which we can establish a cap on the optimal expected accuracy is to test C4.5 on only simple RCC's, and assume that the system should not be able to improve on this performance for coordinated RCC's. This gives a training accuracy of 90.6% and test accuracy of 89.3%, above those for the best-performing *CI_and+vs* configuration. Interestingly, however, the disparity in test accuracies is not statistically significant, such that *CI_and+vs* would appear near optimal.

We see minor performance improvements for the best-performing C4.5 version system over the original system formulation. Looking to the actual rule set inferred from *CI_and+vs* in closed evaluation (with the C4.5 module `c4.5rules`) we see striking similarities with the original system rule set. There are a number of occurrences of low-applicability, high-specificity rules in the C4.5 rule set which were not contained in the original rule set, generally representing over-training on the input data. At the same time, no generalised rule not picked up on in the original system was induced. One interesting effect was that C4.5 was able to apply negative evidence more effectively than the original system formulation in enhancing the precision

[6]Note that, according to the t-test, the training accuracies for *CI_or+vsa* and *FC+vsa* are superior to those for their respective non-VSA counterparts, with confidence $\alpha = 0.001$, and there is no significant difference between the test accuracies for the four basic configurations with and without VSA's.

of various rule instances, and at the same time maintain recall by positing multiple rules founded around the same positive evidence.

One predictable correlation to come from the inferred data, is the high degree of correspondence between agentive head nouns and SUBJECT and CO-ACTOR case-role gapping analyses, and locative head nouns and the various local case-role gapping analysis types. C4.5 was also able to hone in on the true sense of the head noun through complex combinations of head noun semantic parameters.

RCC types the system seemed to have most trouble classifying were BOUND case-role gapping and GENERAL RESTRICTIVE clauses, two clause types which also proved problematic for the original system. In the case of BOUND case-role gapping clauses, for example, seven separate rules were posited to cover only 55 RCC instances, at an average of 7.9 attributes per rule. Even here, C4.5 is only able to achieve a precision and recall of 74.55%, although this does compare favourably against the over-generalised approach adopted in the original research, performing at a precision of 45.9% and recall of 61.8% (under the standard definitions for precision and recall). The heavy-handed methodology adopted in the original research to recognise GENERAL RESTRICTIVE RCC's, was essentially to have a lexicon of head nouns which commonly produce this analysis type. On detection of such "non-gapping" head nouns and non-triggering of any other head restrictive RCC type, we assume the clause to be GENERAL RESTRICTIVE. This coarse technique was carried over to the parameterisation of the data set, and C4.5 appeared unable to fashion any more reliable generalised technique to produce this analysis type.

5.3 Additional evaluation

One item not covered in the original research was the relevant successes and interplay between the different parameter types, in determination of the RCC construal type. We are now in a position to be able to partition off the parameter space and run C4.5 on the different combinations thereof, through the medium of the *CI_and+vs* system configuration. The particular parameter partitions we are interested in are: case slot compatibility flags (C – 11 attributes), head noun semantics (N – 14 attributes) and verb classes (V – 27 attributes). We additionally apply VSA's (VSA – 36 attributes) in isolation to gauge their potential in RCC analysis, and directly compare them to the system of verb classes proposed in the original research.

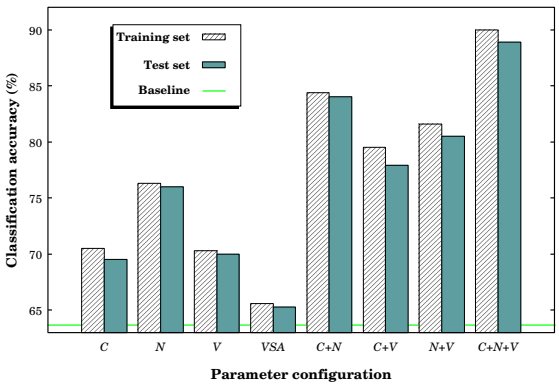

Figure 2: Evaluation of different parameter settings

The system results over the individual parameter partitions, and the various combinations of case slot compatibility, head noun semantics and verb classes (e.g. $N+V$ = head noun semantics and verb classes), are presented in Figure 2.[7] The value of head noun semantics is borne out by the high test accuracy for N of 76.0%. We can additionally see that case slot instantiation and verb class member attributes provide approximately equivalent discriminatory power, both well above the absolute baseline of 63.9%. This is despite case slot instantiation flags being less than half the number of verb classes, largely due to the direct correlation between case slot instantiation judgements and case-role analyses, which account for around 80% of all RCC's. The accuracy for VSA's is well down, just above the absolute baseline accuracy in testing at 65.3%. Verb classes thus provide a clear advantage over VSA's in RCC analysis.

The affinity between case slot instantiation judgements and the semantics of the head noun are evidenced in the strong performance of $C+N$, although even here, verb classes gain us an additional 5% of performance. Essentially what is occurring here is that associational preferences between particular head noun semantics and certain case-roles/analysis types are incrementally enhanced as we add in the extra dimensions of case slot instantiation and verb classes. The crude set of selectional preferences produced for each analysis type by head noun semantics is enhanced by case slot instantiation values, due to the filtering off of case-role gapping analyses where the associated case slot is instantiated or not contained in the case frame for that clause. Subsequently adding in verb classes produces better localisation of the selectional preferences to the different verb types, and at the same time allows for more regulated interaction between particular case slot positions. The orthogonality of the three dimensions is demonstrated by the incremental performance improvement as we add in extra parameter partitions.

Another item worthy of interest is the learning curve for the C4.5 system. Here, we target the *CI_and+vs* system and run it over data sets of 100, 250, 500, 1000, 2000, 3000, 4000, 5000 and finally 5143 RCC instances, with each data set comprising a proper subset of those larger than it. The training and test accuracies over these data sets are given in Figure 3.

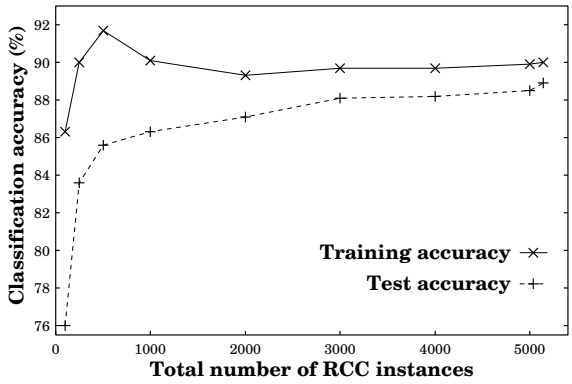

Figure 3: Learning curve

Other than the characteristic knoll at the lower reaches of the training curve, caused by over-training, the training accuracy appears to be levelling out to a figure somewhere between 90 and 91%. The lack of significant diversion beyond about 3000 entries would tend to suggest that our training accuracy is not going to increase much given extra data, and that the 91% accuracy is a ceiling on test performance.

As a final point of evaluation, we randomly extracted a set of 100 fresh RCC's from the EDR corpus to test the robustness of the original system as compared to C4.5. The particular RCC's extracted proved difficult for both systems, with the original system producing an accuracy of 67.0%, as compared to a proportionally deflated 69.0% for the *CI_and+vs* configuration of C4.5. The fact that both systems should have performed equally badly sug-

[7] Note that $C+N+V$ corresponds to the full parameter space (without VSA's), and is identical to *CI_and+vs* in Figure 1.

gests that they are able to handle unseen data equivalently well. From this, we can make the statement that the original system is as robust to new data as could be expected given the composition of the original data, and by extension, the original system has been trained near-optimally on the given data.

6 Discussion

The 89% accuracy achieved in evaluation appears satisfactory given the shallow nature of processing and deliberate avoidance of the use of verb-specific selectional restrictions or pragmatics. At the same time, we recognise that one reason we were able to achieve an accuracy as high as this was that generalised selectional restrictions were being generated for each case-role, particularly with the advent of verb classes. In this respect, it is too strong a statement to say that we have been able to empirically refute the claim that selectional restriction-based semantics govern RCC construal. We have demonstrated, however, that pragmatics are the finishing touch in RCC construal rather than a key limiting factor, and that it is possible to generate a high-accuracy method devoid of pragmatics and with recourse to only a bare minimum of selectional restrictions.

The close performative and structural resemblances between the best system configuration for C4.5 and the original system are personally gratifying, but at the same time formulaicly disturbing. That an industry-standard system should be able to gain little more out of the same data suggests that the original formulation was as good as could be realistically expected, but also points to the limitations of the given parameterisation.

Looking to the future, then, how can we improve system accuracy beyond the suggested performance ceiling of between 90 and 91%? As discussed above, there were various analysis types which C4.5 was unable to do much more with than had been achieved in the original formulation, in particular the GENERAL RESTRICTIVE RCC type. We clearly need to introduce new parameters or devise new techniques to handle this construal type, given that it accounts for over 10% of all RCC's. Matsumoto (1997) provides a hint at where to go from here, in talking of such noun heads 'framing' the verb. It is almost as if different nouns select for certain clause types, in the same way that case slots select for different noun types. The interplay of these forces seems to produce either a case-role gapping or head restrictive reading, suggesting the need for full-on semantic analysis and the notion of verb-specific selectional preferences to capture this effect. This in turn suggests that to extend accuracy beyond that already achieved, we will need to look beyond the lexical realisation of the verb to verb sense.

To surmise, we have compared a tried and proven RCC analysis method against the C4.5 automatic classification system, run over various parameter interpretations. C4.5 was able to produce slightly higher accuracies over the same data, but at the same time evidenced the inherent limitations of the given parameterisation and suggested an absolute performance ceiling for the proposed method of no more than 91%. We further looked at complementing the parameterisation with VSA's, with mixed success, and validated the usefulness of the different parameter types used, as well as the disambiguation techniques developed within the original system.

Acknowledgements

This research would not have been possible without the formidable system resources of the NTT translation communication research group, to whom we are indebted. We would also like to thank Oscar Ortega (TITech) for support with C4.5 and Christoph Neumann (TITech) for his characteristically probing comments on an earlier version of this paper.

References

H. Almuallim, Y. Akiba, and T. Yamazaki. 1994. Two methods for learning ALT-J/E rules from examples and a semantic hierarchy. In *Proc. of the 15th International Conference on Computational Linguistics (COLING '94)*, pages 57–63.

T. Baldwin. 1998. *The Analysis of Japanese Relative Clauses*. Master's thesis, Tokyo Institute of Technology.

EDR, 1995. *EDR Electronic Dictionary Technical Guide*. Japan Electronic Dictionary Research Institute, Ltd. (In Japanese).

S. Ikehara, M. Miyazaki, A. Yokoo, S. Shirai, H. Nakaiwa, K. Ogura, Y. Ooyama, and Y. Hayashi. 1997. *Nihongo Goi Taikei – A Japanese Lexicon*. Iwanami Shoten. 5 volumes. (In Japanese).

K. Inoue. 1976. *Henkei Bunpo to Nihongo*. Tokyo: Taishukan.

E. L. Keenan and B. Comrie. 1977. Noun phrase accessibility and universal grammar. *Linguistic Inquiry*, 8(1):63–99.

Yoshiko Matsumoto. 1997. *Noun Modifying Constructions in Japanese*. John Benjamins.

H. Nakaiwa and S. Ikehara. 1997. A system of verbal semantic attributes in Japanese focused on syntactic correspondence between Japanese and English. *Journal of the Information Processing Society of Japan*, 38(2):215–25. (In Japanese).

H. Nakaiwa, A. Yokoo, and S. Ikehara. 1994. A system of verbal semantic attributes focused on the syntactic correspondence between Japanese and English. In *Proc. of the 15th International Conference on Computational Linguistics (COLING '94)*, pages 672–8.

J.R. Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

M. Silverstein. 1976. Hierarchy of features and ergativity. In R.M.W. Dixon, editor, *Grammatical Categories in Australian Languages*, pages 112–71. Humanities Press.

H. Tanaka. 1996. Decision tree learning algorithm with structured attributes: Application to verbal case frame acquisition. In *Proc. of the 16th International Conference on Computational Linguistics (COLING '96)*, pages 943–8.

H. Teramura. 1975–78. Rentai-shushoku no shintakusu to imi Nos. 1–4. In *Nihongo Nihonbunka 4–7*, pages 71–119, 29–78, 1–35, 1–24. Osaka: Osaka Gaikokugo Daigaku. (In Japanese).

K. Yamamoto and E. Sumita. 1998. Feasibility study for ellipsis resolution in dialogues by machine-learning technique. In *Proc. of the 36th Annual Meeting of the ACL and 17th International Conference on Computational Linguistics (COLING/ACL-98)*, pages 1428–34.