An Application and Evaluation of the *C/NC-value* Approach for the Automatic term Recognition of Multi-Word units in Japanese

Hideki MIMA† Sophia ANANIADOU††

†Dept. of Information Science
University of Tokyo
Hongo 7-3-1, Bunkyo-ku, Tokyo 113-0033 Japan
mima@is.s.u-tokyo.ac.jp

††Dept. of Computer Science, School of Sciences
University of Salford
Salford M5 4WT U.K.
S.Ananiadou@salford.ac.uk

Abstract. Technical terms are important for knowledge mining, especially as vast amounts of multi-lingual documents are available over the Internet. Thus, a domain and language-independent method for term recognition is necessary to automatically recognize terms from Internet documents.

The *C-/NC-value* method is an efficient domain-independent multi-word term recognition method which combines linguistic and statistical knowledge. Although the *C-value/NC-value method* is originally based on the recognition of nested terms in English, our aim is to evaluate the application of the method to other languages and to show its feasibility for multi-language environment.

In this paper, we describe the application of the *C/NC-value* method to Japanese texts. Several experiments analysing the performance of the method using the NACSIS Japanese AI-domain corpus demonstrate that the method can be utilized to realize a practical domain- and language-independent term recognition system.

Keywords: Automatic term recognition, C-value, NC-value, nested terms, term context word

1. Introduction

With the recent dramatic increase of importance of electronic communication and data-sharing over the internet, there is a growing number of publicly accessible global knowledge sources, such as documents.

In general, technical terms represent the most important concepts of a document and characterize the document semantically. In specialized fields, especially in areas such as computer science, biology, and medicine, there is a growing number of new terms that represent newly created concepts. As existing term dictionaries cannot cover the needs of specialists, a domain- and language-independent automatic term recognition (ATR) method is necessary for efficient term discovery (Mima 1998). The *C/NC-value* method is an efficient domain-independent multi-word term recognition method, which combines linguistic and statistical information (Frantzi 2000). In this paper, we apply and evaluate this method which originally extracts multi-word terms from English corpora.

This paper is divided into two parts: in the first part we describe the *C-value* which aims to improve the extraction of nested multi-word terms and collocations (Frantzi 1999) and in the second part, we describe the *NC-value* which incorporates context information into the *C-value* method, thus aiming to improve multi-word term extraction (Frantzi 1999). We also describe a method for the extraction of context words around terms. Although the *C-value/NC-value* method was originally developed for nested terms in English, our aim is to evaluate its application to other languages, such as Japanese and to show its feasibility in a multi-language environment.

In this paper, we primarily explain the *C/NC-value* method as an efficient system for domain-independent term recognition. We then describe an application of the method to Japanese texts.

Since ATR methods are mostly empirical (Kageura 1996), we evaluate the results of the method in terms of precision and recall (Salton 1983). The results are compared with those produced by the most common statistical technique used for ATR to date, the frequency of occurrence of the candidate term, which was applied to the same corpus.

Several experiments analysing the performance of the method using the NACSIS Japanese AI-domain corpus demonstrate that the method can be utilized to realize a practical domain- and language-independent term recognition system.

2. The C-value Approach

In this section we present briefly the *C-value* approach (Frantzi 2000) to multi-word ATR and its application to Japanese, together with a performance evaluation using the NACSIS Japanese corpus. *C-value* is a domain-independent method for multi-word ATR, which aims to improve the extraction of nested terms. The method takes as input a corpus and produces a list of candidate multi-word terms. These are ordered by their *termhood*, i.e. their likelihood of being valid technical terms. The higher the C-value result the greater is the likelihood of a candidate term being a valid term. The *C-value* approach combines linguistic and statistical information, with an emphasis on the statistical part. The linguistic information consists of linguistic filters based on the part-of-speech tagging of the corpus and a stop-list. The statistical part combines statistical features of the candidate string, in the form of a measure, also called *C-value*.

2.1 The Linguistic Part

The linguistic part consists of the following:

1. Part-of-speech information after tagging the corpus.
2. Linguistic filters applied to the tagged corpus to exclude those strings not required for extraction.
3. A stop-list.

Tagging

Part-of-speech tagging is the assignment of a grammatical tag (e.g. noun, adjective, verb, preposition, determiner, etc.) to each word in the corpus. This information is needed by the linguistic filters, which will only permit specific strings for extraction.

The linguistic filter

The purpose of applying the linguistic filter applied to a tagged corpus is to detect possible patterns as terms. However, the choice of the linguistic filter affects the precision and recall of the output list. We have experimented with a number of different filters in English:

1. *Noun+ Noun*
2. *(Adj|Noun)+ Noun*
3. *((Adj|Noun)+ |((Adj|Noun)* (NounPrep)?) (Adj|Noun)*) Noun*

In general, a ‘closed’ filter, which is strict about the strings it permits, will have a positive effect on precision but a negative effect on recall.

Experiments on biological corpora revealed that the addition of prepositions as part of the linguistic filter influenced the precision negatively.

An ‘open’ filter, one that permits several types of strings, has the opposite effect: negative for precision, positive for recall. Therefore, the choice of the linguistic filter depends on how we want to balance precision and recall: preference for precision over recall would probably require a closed filter, while preference for recall would require an open filter. We are not strict about the choice of a specific linguistic filter, since different applications require different filters. We will present our method combined with the filters for Japanese in section 2.4, together with the performance evaluation regarding the choice of linguistic filters.

The stop-list

A stop-list for ATR is a list of words which are not expected to occur as terms in that domain. It is used to avoid the extraction of strings that are unlikely to be terms, thereby improving the precision of the output list.

The stop list can be progressively refined following the initial results of the *C/NC value*. A refined stop list drastically improves precision.

2.2 The Statistical Part

The *C-value* statistical measure assigns *termhood* to a candidate string, placing it in the output list of candidate terms. The measure is based on the following statistical characteristics of the candidate string:

1. The total frequency of occurrence of the candidate string in the corpus.
2. The frequency of the candidate string as part of other longer candidate terms.
3. The number of these longer candidate terms.
4. The length of the candidate string (in number of words).

1) The use of the total frequency of occurrence of the candidate string in the corpus

Frequency generally produces good results since terms tend to occur with relatively high frequencies. However, an advantageous characteristic of the *C-value* approach is to focus not only on the frequencies of candidate terms, but also on the linguistic nature of *nested terms*[1]. For example[2], consider the string *soft contact lens*. A method that uses frequency of occurrence would extract it given that it appears frequently enough in the corpus. Its substrings, *soft contact* and *contact lens*, would also be extracted since they would have frequencies at least as high as *soft contact lens* (and they satisfy the linguistic filter used for the extraction of *soft contact lens*). However, *soft contact* is not a term.

2) The use of the frequency of the candidate string as part of other longer candidate terms

A quick solution to this problem is to extract only a substring of a candidate term if it appears a sufficient number of times by itself in the corpus (i.e. not only as a substring). Then, in order to calculate the *termhood* of a string, we should subtract from its total frequency its frequency as a substring of longer candidate terms

$$termhood(a) = f(a) - \sum_{b \in T_a} f(b) \qquad (1)$$

where,

---

[1] Where we call *nested terms* those that appear within other longer terms, and may or may not appear by themselves in the corpus.

[2] Although, in this section, we use English examples to facilitate understanding the original idea of *nested terms*, it is obvious that the same linguistic phenomenon can also be observed in Japanese from the view point of the growing number of ‘borrowed’ morphemes (Kageura 1998).

a is the candidate string,
$f(a)$ is its total frequency of occurrence in the corpus,
T_a is the set of candidate terms that contain a,
b is such a candidate term,
$f(b)$ is the frequency of the candidate term b that contains a.

3) The use of the number of longer candidate terms that our string appears as nested in

However, the problem is not totally solved using the above measure. Consider the following two sets of terms from computer science.

real time image generation	floating point operation
real time clock	floating point arithmetic
real time expert system	floating point constant
real time image generation	floating point operation
real time output	floating point routine
real time systems	

Both of these two sets contain *nested terms*. The first set contains the term *real time* and the second the term *floating point*. Except *expert system*, all of the other substrings, *time clock*, *time expert system*, *time image generation*, *image generation*, *time output*, *time systems*, *point arithmetic*, *point constant*, *point operation*, *point routine*, are not terms. So substrings of terms may or may not be terms themselves. Also, terms that are substrings do not have to appear by themselves in a text. As a result, a measure like formula (1) would exclude terms if these have been only found as nested, or if they are not nested but present a very low frequency.

The evidence adopted in the *C-value* method to solve the problem is that the higher the number of longer terms that our string appears as nested in, the more certain we can be about its independence. In the above two sets of examples, *real time* appears in every term of the first set, and *floating point* in every term of the second. We have no such evidence for *time clock*, *time expert system*, *time image generation*, *image generation*, *time output*, *time systems*, *point arithmetic*, *point constant*, *point operation*, *point routine*. Because *real time* appears in 5 longer terms, and *floating point* in 4 longer terms, indicates that both show sufficient 'independence' from the longer terms they appear in. This is not the case for *time clock*, which only appears in one term.

4) The length of the candidate string (in number of words)

The last parameter in the *C-value* measure is the length of the candidate string with respect to the number of words. Since it is less probable that a longer string will appear f times in a corpus rather than a shorter string[3], the fact that a longer string appears f times is more important than that

---

[3] This is based on the assumption that the probability of occurrence of the word a in the corpus is independent of the probability of occurrence of any other word in the corpus, which is not always true

of a shorter string appearing f times. For this reason, the length of the candidate string is incorporated into the measure.

Since maximum length terms cannot be nested in longer terms, and some strings are never found as nested, we distinguish two cases

1. If a is a string of maximum length or has not been found as nested, then its *termhood* will be the result of its length and its total frequency in the corpus.
2. If a is a shorter string of any other length, then we must consider if it is part of any longer candidate terms.

If a string appears as part of longer candidate terms, then its *termhood* will also depend on its frequency as a nested string, as well as the number of these longer candidate terms. Despite the fact that appearing as part of longer candidate terms affects its *termhood* negatively, the bigger the number of these candidate terms, the higher is its independence from these. This latter number moderates the negative effect of the candidate string being nested in longer candidate terms.

The measure of *termhood*, called *C-value* is given as

$$C-value(a)=\begin{cases}\log_2|a|\cdot f(a) & a \text{ is not nested,}\\ \log_2|a|\left(f(a)-\frac{1}{P(T_a)}\sum_{b\in T_a}f(b)\right) & \text{otherwise}\end{cases} \qquad (2)$$

Where

a is the candidate string,

$|a|$ is the length of the candidate string a (in number of words),

$f(.)$ is its frequency of occurrence in the corpus,

T_a is the set of extracted candidate terms that contain a,

$P(T_a)$ is the number of these candidate terms.

It is obvious that *C-value* is a measure based on the frequency of occurrence of a. The negative effect on the candidate string a being a substring of other longer candidate terms is marked by the negative sign '-' in front of the total sum of $f(b)$. The independence of a from these longer candidate terms is given by $P(T_a)$. The fact that the greater this number the bigger its independence (and vice versa), is reflected by having $P(T_a)$ as the denominator of a negatively signed fraction. The positive effect of the length of the candidate string is moderated by the application of the logarithm on it.

Further details and examples on how *C-value* works together with the performance evaluation

---

(Dunning 1993).

using an English corpus can be found in (Frantzi 1999), (Frantzi 2000).

2.3 The *C-value* approach to Japanese

In order to apply the *C-value* method to Japanese it is necessary to choose the appropriate linguistic filter(s) for Japanese. As we have already mentioned, the choice of linguistic filter depends on how we want to balance precision and recall. Furthermore, in order to secure positive effect for frequency-based ATR, 'open' filters are preferred. For Japanese, we currently adopt the following patterns as linguistic filters of *termhood* in Japanese.

[Filter 1] *Noun{2,}*

Ex. "*giji-raNsûseisû-keiretsu" (pseudo-random numbers)*[4], *"kasou-kansû" (virtual function)*

[Filter 2] *(Prefix | Adv) (Noun | Adj | Suffix)+ Noun+*

Ex. *"zen-nijyû-setsuzoku" (full duplex connection), "hi-douki-sûshiN" (asynchronous transmission)*

[Filter 3] *Prefix Noun+ Suffix*

Ex. *"mi-teigi-kata" (undefined type), "sai-syoki-ka" (re-initialize)*

However, in order to make a thorough extraction down to the minutest candidate terms for Japanese, the following issues must be resolved.

- **How can we deal with borrowed terms (Katakana strings[5]) and unknown words?**

Since Japanese is an agglutinating language, it is necessary, to segment it into appropriate morphemes[6] initially by using a morpheme dictionary. Research has shown that, in Japanese, borrowed terms are growing in number (Kageura 1998), making the maintenance of morpheme dictionaries difficult. Thus, it is a rather common feature of morphological analysers to recognize Katakana strings not as morphemes but as unknown words (Kurohashi 1998). As a result, we are not able to obtain the necessary information from morpheme segments (part-of-speech information) of every borrowed term.

Although these errors might not affect the performance of certain techniques, i.e. calculating the frequency of occurrence, they have to be dealt with in other cases. (Hisamitsu 1998) deals with the problem of segmentation errors in Japanese by post-processing, i.e. using transformation rules and contextual information.

- **How can we retrieve nested collocations in borrowed terms?**

As *C-value* is based on the recognition of nested terms, it requires minimum morpheme

---

[4] In this paper, sample Japanese is Romanized in italic based on the Hepburn system with the corresponding English words following in parentheses.

[5] Although not every Katakana string in Japanese is a borrowed word, in this paper, we use the word 'Katakana' string with the same meaning as borrowed word.

[6] In this paper, we use the word 'morpheme' as the linguistic representation of the smallest meaningful element.

recognition. The recognition of nested terms requires morpheme string matching.

However, as mentioned above, since we may not be able to obtain original segments from borrowed terms, we cannot expect to find any morpheme segments in the borrowed terms. As a consequence, the retrieval of nested collocations may fail.

In order to solve the above problems, the following assumptions are taken into account in the linguistic filters and the *C-value*:

- **Allow Katakana strings (unknown words) to be substituted into *Nouns* in the linguistic filters.**

  Ex. *"kyôyû-memori" (shared memory), "tyokusetsu-kakusaN-supekutoramu-tsûshiN-hôshiki" (direct sequence spread spectrum transmission)*
- **Perform character level string matching to every Katakana string.**

  To extend the coverage for detecting the nested terms in Japanese, we use character level string pattern matching instead of using word (morpheme) level pattern matching for Katakana strings.

2.5 Evaluation

We have conducted experiments to examine the performance of the method with respect to:

1) The precision of the top 10% of the resulting list including the individual score for nested terms according to each linguistic filter.
2) The overall performance from the point of view of precision and the recall by 11-point score[7].
3) The interval value of the precision to confirm the 'practical' performance, while applying it to Japanese texts.

The results are based on both tagged and untagged corpora. We use JUMAN (Kurohashi 1998) for the morphological analysis of Japanese in order to tag the corpus which is used for the evaluation of the untagged corpus. We used the NACSIS tagged/untagged corpus (Koyama 1998) as input and evaluated the results by comparing the TMREC Manual- and Index-Candidates as a correction set (Kageura 1999). In the evaluation, we only observed the tendencies on the basis of exact match, i.e. with respect to the complete coincidence with the correction set. As we mentioned earlier, ATR techniques are mostly based on frequency of occurrence, on the assumption that terms tend to appear with high frequency. We have therefore evaluated the results of the C-value method in terms of precision and recall and compared it with frequency of occurrence.

In table 1, we show, the corresponding precision for each of the three linguistic filters and for the whole result, and we compare them with the corresponding results of frequency of occurrence. Since *C-value* is a method which aims to improve the extraction of nested terms, the comparison

---

[7] Where 11-point score indicates that, for example, precision at recall 0.10 is taken to be maximum of precision at all recall points ≥ 0.10.

investigated its advantage by showing the results for individual precision of nested terms. In the table, the label "nested terms" indicates the candidate terms that have also appeared as nested (i.e. the candidate terms that have also appeared by themselves in the corpus), the label "all" indicates all the candidate terms extracted by the *C-value* and by the frequency of occurrence, using the corresponding linguistic filters. We used the results of the top 10% of the list for the evaluation. As for the frequency of occurrence, we evaluated the same number of the results. The results show that we can obtain high precision for "nested terms" i.e. all results of *C-value* and frequency of occurrence, indicating that making use of the linguistic information, i.e. nested terms, affects the performance positively, even for Japanese texts.

On the other hand, regarding the selection of the linguistic filters, although the results show that in Japanese almost all the candidate terms are derived from filter 1, i.e. *Noun{2,}* we nevertheless observed substantial differences in the precision for the combinations of the three linguistic filters. However, this also indicates that we can expect that this result strengthens the argument that by using *C-value* we have the freedom to use a more open linguistic filter.

In figure 1 we calculate the simple 11-point score[8] applied to the results of the *C-value* method using both tagged and untagged corpora, together with those of frequency of occurrence. We calculated the results with a weight greater than 1.0 for the evaluation of *C-value* and frequency of occurrence. The numbers of the candidate terms produced by the *C-value* method for tagged and untagged corpora are 9548 and 9815 respectively. For frequency of occurrence for tagged and untagged corpora the results are 10403 and 10574.

We also provide, in figures 2 and 3 the results of the fixed interval precision[9] of *C-value* and frequency of occurrence with different corpora, i.e. tagged and untagged. In figures 2 and 3, we observe that *C-value* increases the concentration of real terms at the top of the list. More precisely, we observe that *C-value* increases the precision for the first 3 intervals by 5 %. It is noteworthy that the precision of the top 100 intervals in both cases is more than 93 %, which is high in quality.

The above results show that *C-value* produces more real terms than pure frequency of occurrence, placing them closer to the top of the extracted list, even for Japanese.

The major difference in the performance between tagged and untagged corpora was originally derived from the difference in the results of the part-of-speech tagging, i.e. in the lists of linguistic

---

[8] Readers should note that, 1) because the score of 11-point automatically becomes 0 when the horizontal axis goes over the original recall of the results, it totally depends on the overall recall, however, 2) since the *C-value* method focuses on extracting multi-word terms, not single-word terms, and since single-word terms are also included in the correction set (Koyama 1998), (Kageura 1999), they are not the absolute scores for recall to be compare with the other single-word related ATR methods.

[9] Where the fixed interval precision indicates a series of precisions for certain fixed intervals from the top of the candidate term list, e.g. the precisions of intervals 1-100, 101-200, 201-300, etc. from the top of the list.

filters. Therefore, further selection of linguistic filters is thought to be required for tagged corpora to obtain the corresponding score. On the other hand, this also shows that we can always expect high performance even for a 'practical' use of the method.

Table 1. Precision: *C-value* vs frequency with num. of extracted terms

		filter 1	filter 1+2	filter 1+2+3
C-value nested terms	tagged untagged	86.8 % (385) 90.3 % (371)	87.5 % (399) 89.9 % (386)	87.5 % (400) 89.9 % (387)
C-value all	tagged untagged	77.2 % (1010) 80.8 % (1037)	77.7 % (1028) 81.1 % (1050)	77.4 % (1040) 81.2 % (1057)
frequency all	tagged untagged	67.5 % (1010) 67.7 % (1037)	67.9 % (1028) 67.5 % (1050)	67.3 % (1040) 67.6 % (1057)

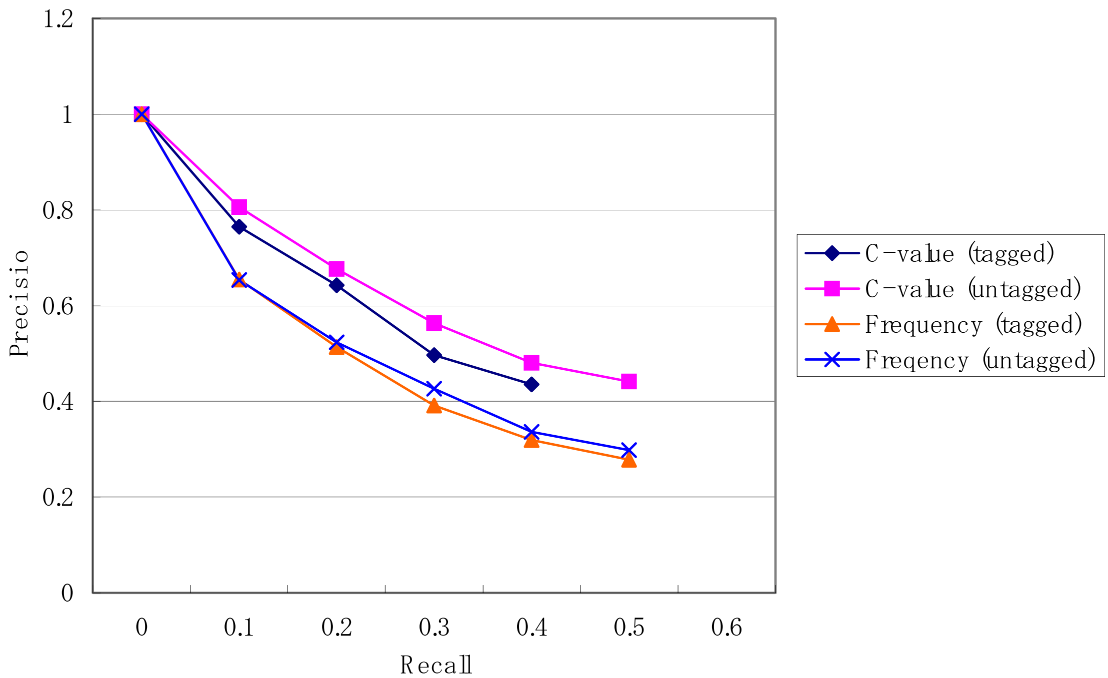

Figure 1. 11-point score of precision and recall

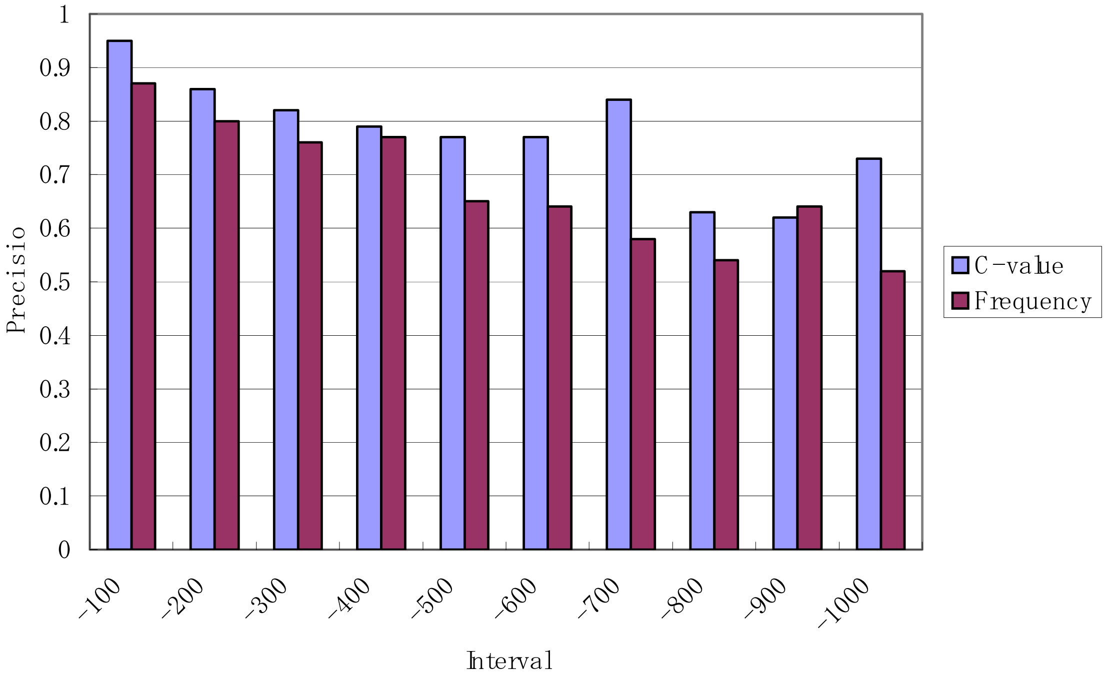

Figure 2. Interval precision: *C-value* vs frequency (tagged corpus)

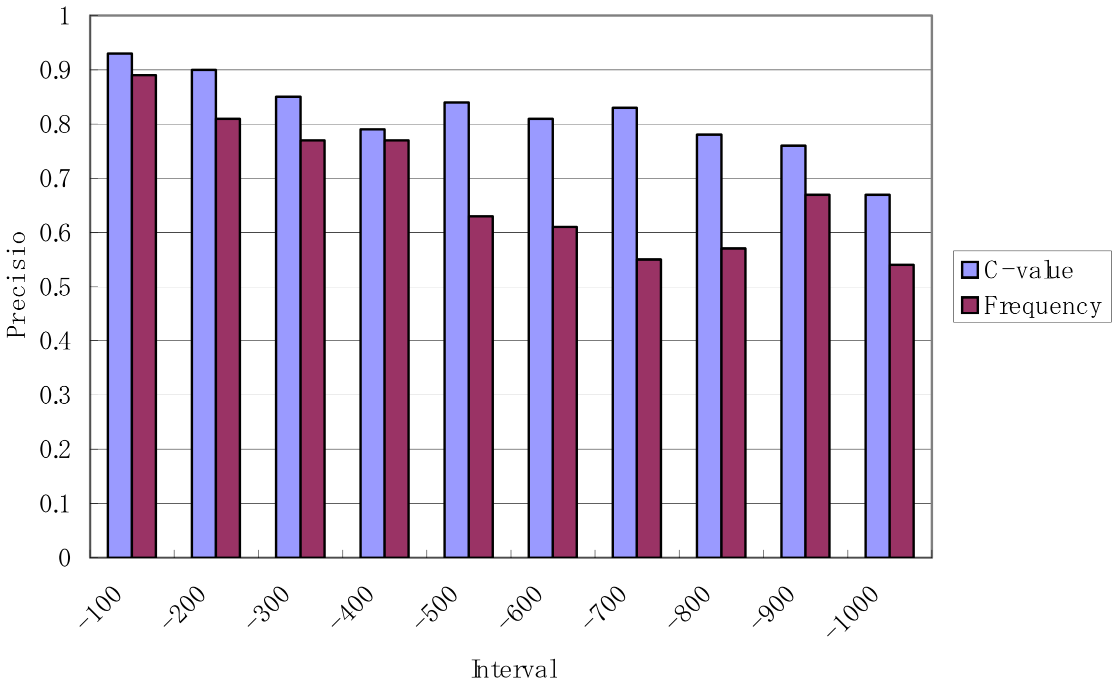

Figure 3. Interval precision: *C-value* vs frequency (untagged corpus)

3. Incorporating Context Information: The *NC-value* Approach

In this section, we briefly describe the NC-value method which incorporates context information into ATR. We often use the environment of a word to identify its meaning. The NC-value approach uses this information on the assumption that, since extended term units differ from extended word units as far as modification is concerned, we could use information from the modifiers to distinguish between terms and non-terms. Thus, for example, if *consistent* is an adjective that tends to precede terms in medical corpora, and it occurs before a candidate term string, we could exploit this information for the benefit of term recognition. Besides adjectives and nouns, we can expand the use of modifier types to verbs that belong to the environment of the candidate term: the string *show* of the verb *to show* in medical domains is often followed by a term, e.g. *shows a basal cell carcinoma*. We will use the three part-of-speech elements also used by (Grefenstette 1994) to obtain information about the *termhood* of a candidate string, when they either precede or follow it. These are

1. nouns (*compound cellular naevus*),
2. adjectives (*blood vessels are present*),
3. verbs (*composed of basaloid papillae*).

3.1 The Context Weighting Factor

In this section we describe a method to create a list of 'important' *term context words* from a set of terms extracted from a specialised corpus. By term context words we mean those that appear in the vicinity of terms in texts. These will be ranked according to their 'importance' when appearing with terms. The context words we treat are adjectives, nouns and verbs that either precede or follow the candidate term.

The criterion for the extraction of a word as a term context word is the number of terms it appears with. The assumption is that the higher this number, the higher the likelihood that the word is 'related' to terms, and that it will occur with other terms in the same corpus. Term context words for a specific domain/corpus are not necessarily the same for another domain/corpus. For this reason, we relate term context words to a specific corpus. For example, the words *present, shows, appear, composed* tend to appear with terms in our medical corpus, but may have a different significance if found in a different domain, e.g. *mathematics*.

We can express the above criterion more formally with the measure

$$weight(w) = \frac{T(w)}{n} \tag{3}$$

where

w is the context word (noun, verb or adjective) to be assigned a weight as a term context word,

Weight(w) the assigned weight to the word w, *t(w)* the number of terms the word *w* appears with,

n the total number of terms considered.

The purpose of the denominator *n* is to express this weight as a probability: the probability that the word *w* might be a term context word.

3.2 *NC-value*

In this subsection we present the method we call *NC-value*, which incorporates context information into the *C-value* method for the extraction of multi-word terms. Assuming we have a corpus from which we want to extract the terms, the *NC-value* algorithm consists of the following three stages

First stage

We apply the *C-value* method to the corpus. The output of this process is a list of candidate terms, ordered by their *C-value*.

Second stage

This involves the extraction of the term context words and their weights. These will be used in the third stage to improve the term distribution in the extracted list. In order to extract the term context words, we need a set of terms, as discussed in the previous section. We have chosen to keep the method domain-independent and fully-automatic (until the manual evaluation of the final list of candidate terms by the domain-expert). Therefore, we do not use any external source (e.g. a dictionary) that will provide us with the set of terms to be used for this purpose. We use instead the 'top' candidate terms from the *C-value* list, which present very high precision on real terms. We expect to find non-terms among these candidate terms that could produce 'noise', but these non-terms are rare enough not to cause any real problems. We have chosen to accept a small amount of noise, i.e. non-terms, for the sake of full automation. These 'top' terms produce a list of term context words and assign to each of them a weight following the process described in the previous section.

Third stage

This involves the incorporation of context information acquired from the second stage of the extraction of multi-word terms. The *C-value* list of candidate terms extracted during stage one is re-ranked using context information, so that the real terms appear closer to the top of the list than they did before, i.e. the concentration of real terms at the top of the list increases while the concentration of those at the bottom decreases. The re-ranking takes place in the following way: Each candidate term from the *C-value* list appears in the corpus with a set of context words. From these context words, we retain the nouns, adjectives and verbs for each candidate term. These words may or may not have been met before, during the second stage of the creation of the list with the term context words. In the case where they have been met, they retain their assigned weight. Otherwise, they are assigned zero weight. For each candidate term, we obtain the context factor by summing up: the weights for its term context words, multiplied by the frequency with which they appear with this candidate term.

For example, assume that the candidate word *W* appears 10 times with the context word c_1, 20 times with the context word c_2, and 30 times with the context word c_3. Assume also that the weight for c_1 is w_1, the weight for c_2 is w_2, and the weight for c_3 is w_3. Then, the context factor for *W* is:

$$10 \cdot w_1 + 20 \cdot w_2 + 30 \cdot w_3$$

The above description is the second factor of the *NC-value* measure which re-ranks the *C-value* list of candidate terms. The first factor is the *C-value* of the candidate terms. The whole *NC-value* measure is formally described as

$$NC\text{-}value(a) = 0.8 * C\text{-}value(a) + 0.2 * CF(a)$$

Where

a is the candidate term,

C-value(a) is *the C-value* for the candidate term *a*,

CF(a) is the context factor for the candidate term.

The two factors of *NC-value*, i.e. *C-value* and the context information factor, have been assigned the weights 0.8 and 0.2 respectively, which have been chosen empirically.

Further details on how *NC-value* works together with the performance evaluation using an English corpus can also be found in (Frantzi 1999), (Frantzi 2000).

3.3 Evaluation

We have also conducted experiments to examine the performance of the *NC-value* method with respect to the overall performance from the viewpoint of precision and recall by 11-point score, while applying it to the same corpus and the correction set to the *C-value*. All the results are given using both tagged and untagged corpora. In the evaluation, we also observed the tendencies on the basis of exact match. The top of the list produced by *C-value* was used[10] for the extraction of term context words[11], since these show high precision on real terms. It is expected that among those terms there will be some non-terms as well. This is unavoidable since we have chosen to keep this process fully-automatic.

The weights 0.8 and 0.2 assigned to *C-value* and the context factor in the *NC-value* measure were also chosen after a series of experiments. As reported in (Frantzi 2000), we also adopted the combination 0.8—0.2, since this combination gave the best distribution in the precision of extracted terms for Japanese as well.

As already mentioned, readers should note that, since *NC-value* re-ranks the *C-value* list

---

[10] The first 20% of the extracted candidate terms were used for these experiments.

[11] We used 30 context words for all the extracted terms in the evaluation, which was also determined empirically.

without adding or deleting any candidate terms, the total recall and the total precision of the *NC-value* are the same as those of the *C-value* list. What is different is the distribution of terms in the extracted list.

Figure 4 and 5 show the 11-point precision-recall score of *NC-value* method in comparison with the corresponding *C-value* in cases of using tagged and untagged corpora, respectively.

In figure 6 and 7, we present the fixed interval precision of the *NC-value* method in comparison with the corresponding *C-value* and frequency of occurrences in cases of using tagged and untagged corpus, respectively.

In the figures for the 11-point precision-recall score, we observe that the *NC-value* increases the precision with compared to that of *C-value* on all the correspond points for recall. More precisely, we observe that *NC-value* increased average precision by 2-3% for the recall of 01-0.4.

For the interval precisions, although we see a small drop in precision compared to *C-value* in some intervals, we observe that *NC-value* generally increases the concentration of real terms at the top of the list.

Thus, from the results, we can expect that the *NC-value* produces more real terms than *C-value*, placing them closer to the top of the extracted list.

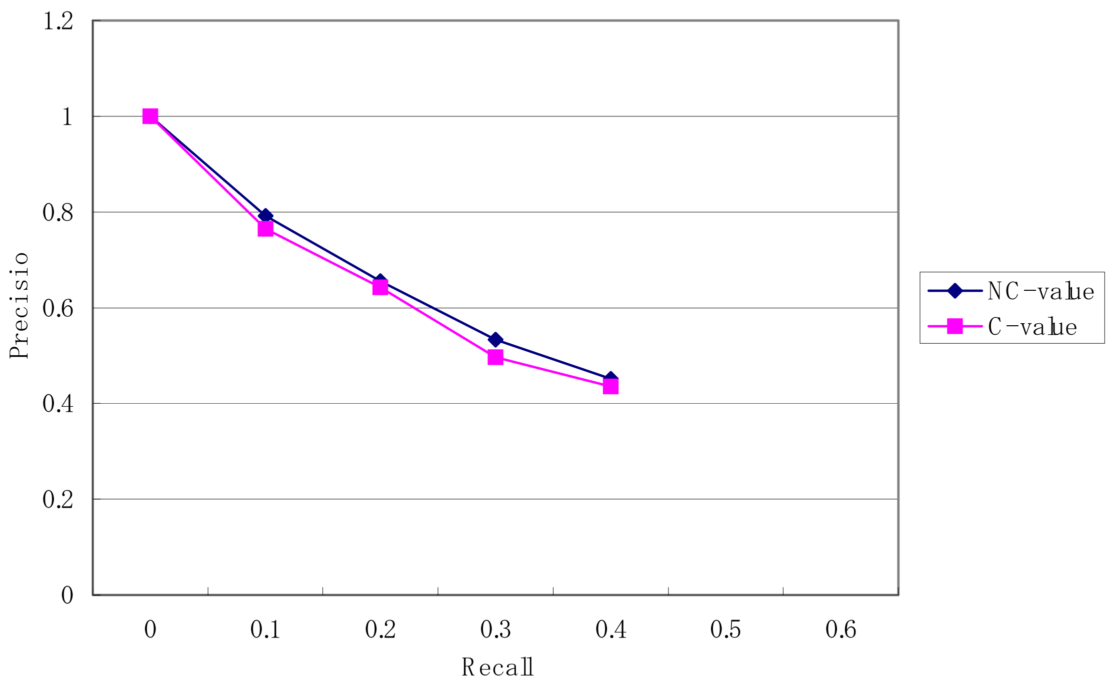

Figure 4. 11-point score: *NC-value* vs *C-value* (tagged corpus)

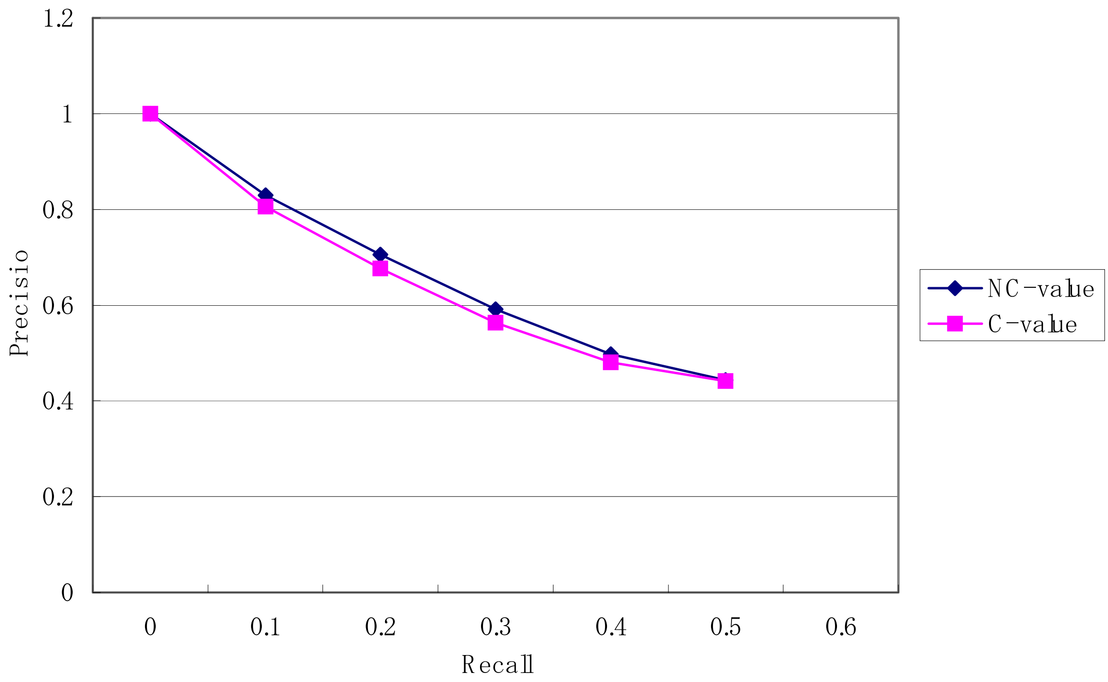

Figure 5. 11-point score: *NC-value* vs *C-value* (untagged corpus)

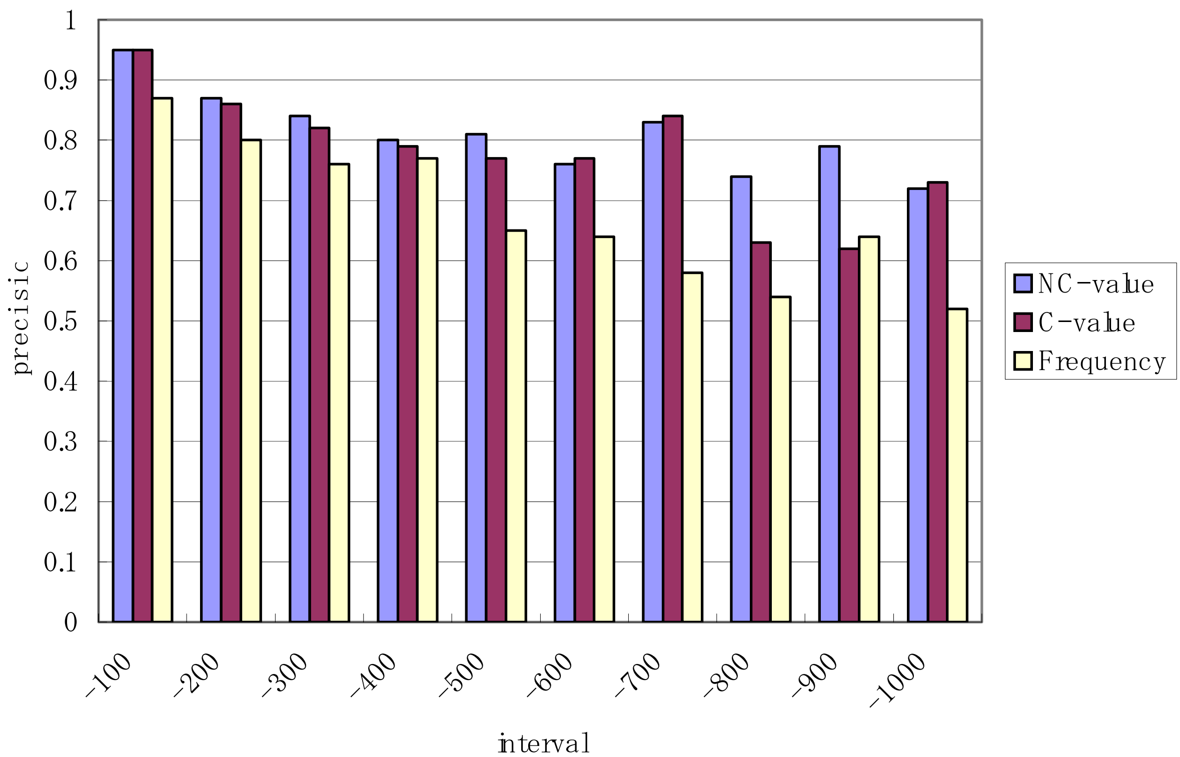

Figure 6. Interval precision: *NC-value* vs *C-value* (tagged corpus)

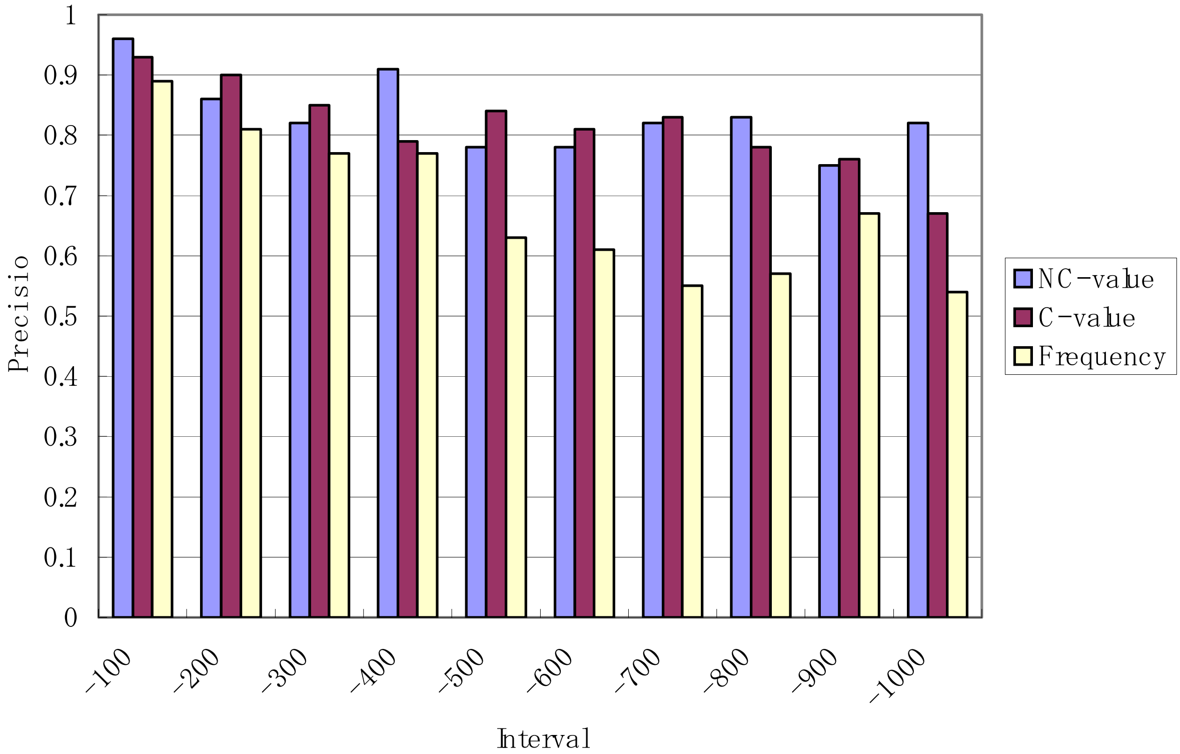

Figure 7. Interval precision: *NC-value* vs *C-value* (untagged corpus)

4. Conclusions

In this paper we presented an application of the *C-value/NC-value* method to Japanese as an efficient domain-independent multi-word term recognition method. We evaluated the method using the NACIS Japanese AI-domain corpus (Koyama 1998).

Although the *C-value/NC-value* method was originally used for the recognition of English nested terms, we demonstrated that both methods are also effective for Japanese term recognition.

Several experiments analysing the performance of these methods using the corpus lead us believe that the schemes can be utilized to realize a practical domain- and language-independent term recognition system.

Important areas of future research will involve:

- Selecting more appropriate linguistic filter(s) to improve the recall in candidate term detection.
- Utilizing semantic-oriented information, such as domain specific thesauri and statistically clustered terms (Ushioda 1996), (Maynard 2000) for improving the performance of term recognition.

Developing a web-based term-oriented knowledge mining system relevant for scientific database information (Mima 1999) to show its practicality in language independent environments is another area of interest for future work.

References

Dunning, T. 1993. "Accurate Methods for the Statistics of Surprise and Coincidence". *Computational Linguistics* 19(1), 61—74.

Frantzi, K. T., Ananiadou, S. and Mima, H. 2000. "Automatic Recognition of Multi-Word Terms: the *C-value/NC-value* method".*International Journal on Digital Libraries* Vol. 3, No. 2, 115—130.

Frantzi, K. T. and Ananiadou, S. 1999. "The C-value/NC-value domain-independent method for multi-word term extraction". *Journal of Natural Language Processing* Vol. 6, No. 3, 145—179.

Grefenstette, G. 1994. "Explorations in Automatic Thesaurus Discovery". *Kluwer Academic Publishers*.

Hisamitsu, T. and Niwa, Y. 1998. "Post-processing of Japanese Morphological Analysis Using Transformation Rules and Contextual Information". *Information Processing Society of Japan (IPSJ) SIG Notes, NLP*, Vol. 98, No. 63, 55—62. (in Japanese).

Kageura, K., Yoshioka, M., Tsuji, K., Yoshikane, F., Takeuchi, K. and Koyama, T. 1999. "Evaluation of the Term Recognition Task". In Proceedings of *the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition*, 417—434. Aug 30-Sep 1, 1999, Tokyo.

Kageura, K. 1998. "A Statistical Analysis of Morphemes in Japanese Terminology". In Proceedings of *The 17th International Conference on Computational Linguistics and The 36th Annual Meeting of the Association for Computational Linguistics, COLING-ACL'98*, 638—645. Montreal, Canada.

Kageura, K. and Umino, B. 1996. "Methods of Automatic Term Recognition -A Review-". *Terminology* 3(2), 259—289.

Kurohashi, S. and Nagao, M. 1998. "Japanese Morphological Analysis System JUMAN". Kyoto University.

Koyama, T., Yoshioka, M. and Kageura, K. 1998. "The Construction of a Lexically Motivated Corpus --- The Problem with Defining Lexical Units ---". In Proceedings of *First International Conference on Language Resources and Evaluation*, 1015—1019. Granada, Spain.

Maynard, D. and Ananiadou, S. 2000. "Identifying Terms by their Family and Friends", In Proceedings of *The 18th International Conference on Computational Linguistics, COLING 2000*, 530—536.

Maynard, D. and Ananiadou, S. Forthcoming. "Trucks: a model for automatic multi-word term recognition". To appear in *Journal of Natural Language Processing*.

Mima, H., Ananiadou, S. and Tsujii, J. 1999. “A Web-based integrated knowledge mining aid system using term-oriented natural language processing”. In Proceedings of *The 5th Natural Language Processing Pacific Rim Symposium*, *NLPRS'99*, 13—18.

Mima, H. Frantzi, K. T. and Ananiadou, S. 1998. “The C-value/Example-based Approach to the Automatic Recognition of Multi-Word Terms for Cross-Language Terminology”. In Proceedings of *The Pacific Rim International Conferences on Artificial Intelligence*, *PRICAI'98*, Joint Workshop on Cross Language Issues in Artificial Intelligence & Issues of Cross Cultural Communication, 10—21. Singapore.

Salton, G. 1983. “Introduction to modern information retrieval”. *Computer Science*. McGraw-Hill.

Ushioda, A. 1996. “Hierarchical Clustering of Words”, In Proceedings of *The 16th International Conference on Computational Linguistics, COLING’96*, 1159—1162. Copenhagen.