Reviewing and Evaluating Automatic Term Recognition Techniques

Ioannis Korkontzelos, Ioannis P. Klapaftis, and Suresh Manandhar

Department of Computer Science, The University of York
Heslington, York, YO10 5NG, UK
{johnkork,giannis,suresh}@cs.york.ac.uk

Abstract. Automatic Term Recognition (*ATR*) is defined as the task of identifying domain specific terms from technical corpora. *Termhood-based* approaches measure the degree that a candidate term refers to a domain specific concept. *Unithood-based* approaches measure the attachment strength of a candidate term constituents. These methods have been evaluated using different, often incompatible evaluation schemes and datasets. This paper provides an overview and a thorough evaluation of state-of-the-art *ATR* methods, under a common evaluation framework, i.e. corpora and evaluation method. Our contributions are two-fold: (1) We compare a number of different *ATR* methods, showing that *termhood-based* methods achieve in general superior performance. (2) We show that the number of independent occurrences of a candidate term is the most effective source for estimating term nestedness, improving *ATR* performance.

Keywords: automatic term recognition, ATR, term extraction.

Introduction

A terminology bank (vocabulary) contains the terms, which refer to the concepts of a domain. Constructing such a vocabulary is crucial, because it is the starting point for many applications such as machine translation, indexing, and ontology learning [8]. Manual construction is time-consuming, error-prone, labour-intensive and unable to deal with the rapid growth of technical terms. *ATR* targets at solving these obstacles.

ATR techniques can be divided into two broad categories: *unithood-based* and *termhood-based* ones [8]. *Unithood* refers to the attachment strength of the constituents of a candidate term. *Termhood* refers to the degree that a candidate term is related to a domain-specific concept. For example, in an eye-pathology corpus, *"soft contact lens"* is a valid term, which has both high *termhood* and *unithood*. However, its frequently occurring substring *"soft contact"*, has high *unithood* and low *termhood*, since it does not refer to a key domain concept.

Unithood-based methods, such as *t-test*, χ^2*-test*, *Log-likelihood* (*LL*) [3] and *pointwise mutual information (PMI)* [1], have been thoroughly evaluated for the task of collocation extraction [3,4,2,14]. In [3,4] the authors show that *LL*

A. Ranta, B. Nordström (Eds.): GoTAL 2008, LNAI 5221, pp. 248–259, 2008.

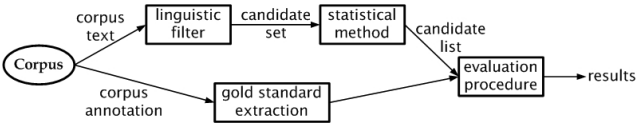

Fig. 1. Experimental procedure

performs better than the other statistical measures due to its milder tendency to overestimate rare events.

Given that *unithood-* and *termhood-based* methods capture different types of information, it is still unclear whether the former are able to perform better than the latter methods, such as *C Value* [5] and *Statistical Barrier* (*SB*) [13]. Furthermore, most *ATR* methods [5,3,1,13] have been evaluated using different technical corpora, under different evaluation frameworks, with different sets of parameters depending on the domain and test corpus. This lack of a common evaluation scheme complicates the interpretation of results. It is unclear which are the strengths and weaknesses of each method, making unmanageable the choice of an appropriate *ATR* method as a starting point for other applications.

This paper provides an overview of the field of *ATR* and evaluates a number of linguistic and statistical approaches using two English corpora i.e. the *GENIA*[6] and the *PennBioIE* [9] corpus. Figure 1 presents a block diagram of our experimental procedure. A linguistic filter is applied on the corpus text to identify candidate terms. Then, a statistical method ranks these candidates, to create a list in decreasing order of scores. The evaluation scheme compares this list to the gold standard terms, generated by the corpus annotation. The scheme consists of a manually annotated corpus, and an evaluation method which assesses the performance of *ATR* methods at a fine-grained scale; i.e. increments of 0.5% of their candidate term ranked list, based on the one proposed in [16].

Our contributions are two-fold: firstly, we extensively compare state-of-the-art approaches to *ATR* under a common evaluation scheme. We show that *termhood-based* approaches, which take into consideration the nestedness of a candidate term into others, such as *C Value* and *SB*, have in general superior performance over methods which measure the strength of association among the tokens of a multi-word candidate term, such as *LL* and *PMI*. Secondly, after further experimentation with different statistical approaches to nestedness we show that the independent occurrences[1] of a term is the most effective source of nestedness information, clearly improving the performance of ATR methods, in this evaluation setting.

The rest of the paper is structured as follows: Sections 1 and 2 review linguistic filtering and statistical approaches, respectively. Section 3 presents the evaluation scheme, the experimental results and comments on them. Section 4 concludes this paper.

[1] Number of occurrences on its own; without being nested within others candidate terms.

1 Linguistic Filters

Initial *ATR* research focused on exploiting the parts-of-speech (*PoS*) of multi-word expression constituents. As a result, different pattern-based models (linguistic filters) were proposed, to identify terms. For example, the linguistic filter in formula 2 would recognise terms consisting of nouns (N) or adjectives (A). The choice of linguistic filter depends on the language and the domain of the corpus and the application [5]. If the target is to identify terms with high recall an open filter should be used, such as the one in formula 3, which applies on numbers ($\#$) and prepositions (P).

In this paper, four lenient *PoS* filters were employed to capture as many terms as possible. Their performance was experimentally compared. The most basic, *Nouns*, accepts sequences of *Ns*, only, since terms mainly consist of *Ns*. The second, *A&N*, applies on sequences consisting of *As* and *Ns* ending with a N (formula 1). The third linguistic filter, *J&K* (formula 2) was introduced by Justeson and Katz [7] and has been widely used. Its first part is identical to *A&N*, whereas the second applies on sequences which start with one or more *Ns* or *As*, continue with a N followed by a P and end with zero or more *Ns* or *As* followed by a N. Justeson and Katz [7] used this filter to extract multi-word terms from large text collections in a variety of domains -metallurgy, space engineering and nuclear energy-, reporting coverage of 97% (99% if *Ps* are allowed).

$$(A|N)^+ \; N \tag{1}$$

$$(\; (A|N)^+ \mid (A|N)^* \; (NP)? \; (A|N)^* \;) \; N \tag{2}$$

$$(\; (A|N|\#)^+ \mid (A|N|\#)^* \; (NP)? \; (A|N|\#)^* \;) \; N \tag{3}$$

Nouns and *A&N* extract sequences of *As*, *Ps* and *Ns*. However, our initial experimental projections show that approximately 6% of *GENIA* gold standard terms contain numbers. To capture those, we extended *J&K* to *J&K#* (formula 3), so as to accept numbers ($\#$) whenever it accepts *Ns* or *As*.

2 *ATR* Statistical Approaches

Approaches to *ATR* have been largely based on statistical information. However, most of them include some linguistic part; usually a linguistic filter, to produce a list of candidate terms (section 1). The statistical part assigns to each candidate term, *ct*, a score, indicating how likely *ct* is a valid term. The most simple statistical measure is the *frequency of occurrence* (*FR*), which captures terms occurring frequently in the corpus. *FR* is used as a baseline in our evaluation.

2.1 Termhood-Based Methods

C Value [5] focuses on nested terms. The basic intuition is that a candidate term, *ct*, should occur frequently on its own, not nested in other candidate terms. For example, in an eye-pathology corpus, *"soft contact lens"* is a valid term, possibly

occurring frequently. However, its substring *"soft contact"* is not an actual term and should not be extracted, since it occurs frequently as nested [5].

However, the nested frequency of *ct* is not a reliable measure of its nestedness, since it does not take into account the number of different candidate terms, in which *ct* appears as nested. For example, consider the following terms in the domain of real time systems: *"real time clock"*, *"real time systems"*, *"real time group"* and *"real time expert system"*. The fact that they all contain *"real time"* as substring, increases its possibility to be a term.

Consequently, the nestedness, NST, of *ct* is defined as the fraction of its nested frequency over the number of distinct candidate terms, in which it appears as nested. The length of a *ct* in tokens, $|ct|$, is also taken into account. The longer *ct* is, the more likely *ct* is an actual term.

$$NST(ct) = \frac{1}{P(T_{ct})} * \sum_{b \in T_{ct}} f(b)) \tag{4}$$

In order to compute a *termhood* value, Frantzi et al. [5] subtract the nestedness, NST, of *ct* from its *frequency of occurrence*, $f(ct)$. In case that *ct* appears as nested, *C Value* is defined by the upper branch of equation 5, where T_{ct} is the set of candidate terms, in which *ct* appears as nested, $P(T_{ct})$ is its cardinality and $L(ct) = log_2(|ct|)$. In the opposite case, *ct* is assigned a value based on its length and *frequency of occurrence* (lower branch of equation 5).

$$CV(ct) = \begin{cases} (f(ct) - NST(ct))L(ct), \text{ nested } ct \\ f(ct)L(ct), \text{ otherwise} \end{cases} \tag{5}$$

NC Value incorporates contextual information into the *C Value ATR* process. It consists of three parts. Firstly, *C Value* is applied on a corpus *cp*, to extract a ranked list of candidate terms, *l*. Secondly, the top *n* candidate terms are selected from *l*. For each of these, its context words *cw* are collected, using a window of $\pm w$ words around it. Context words can be nouns, adjectives or verbs. For each *cw*, the following weight is computed as: $w(cw) = \frac{t(cw)}{n}$, where $t(w)$ is the number of candidate terms *cw* appears with.

Thirdly, the *C Value* ranked list is refined by applying the weights $w(cw)$ to compute a context factor, CF, for each *ct*. The context factor of a $ct \in l$ is formally defined by equation 6, where C_{ct} is the set of context words of *ct*, *b* is an element of C_{ct}, $f_{ct}(b)$ is its *frequency of occurrence* as a context word and $w(b)$ is its weight as a context word. In the case that *b* was not encountered during the stage of creating the list of context words it is assigned a 0 weight. *NC Value* is computed as the linear interpolation of *C Value (CV)* and CF (equation 7).

$$CF(ct) = \sum_{b \in C_{ct}} f_{ct}(b) * w(b) \tag{6}$$

$$NCV(ct) = 0.8 * CV(ct) + 0.2 * CF(ct) \tag{7}$$

Statistical Barrier (SB) [13] is another *ATR termhood-based* approach, which assumes that terms having complex structure are made of existing simple terms.

Thus, they first measure the *termhood* of single words, and then use it to measure the *termhood* of complex terms. The basic intuition is that if a single word N, expresses a key concept of a domain, then N occurs not only frequently, but also in various ways. Thus, there will be a number of valid terms containing N. This potential relationship between single words and multi-word candidate terms is exploited to perform *ATR*.

In particular, after *PoS* tagging a given corpus, Nakagawa [13] extracts a list of single words. Let $R(N)$ and $S(N)$ be two functions that calculate the number of distinct words that adjoin N or N adjoins, respectively. Then, for each candidate term, $ct = N_1, N_2, \ldots, N_k$ a score is calculated (equation 8).

$$IMP(ct) = (\prod_{i=1}^{k}((R(N_i) + 1) * (S(N_i) + 1)))^{1/2k} \tag{8}$$

Nakagawa [13] notes that the frequency of independent occurrences of candidate terms have a significant impact on the term recognition process. Independent occurrences are the ones, where the candidate term ct, is not nested to any other candidate term. To incorporate this, *IMP* is multiplied by the *marginal frequency*, $MF(ct)$, the number of independent occurrences of ct (equation 9).

$$SB(ct) = IMP(ct)MF(ct) \tag{9}$$

2.2 Unithood-Based Methods

Termhood-based methods focus on measuring how likely a candidate term, ct, is a domain-specific concept, by considering nestedness information. On the contrary, *unithood-based* methods attempt to identify if the constituents of a multi-word candidate term form a collocation rather than co-occurring by chance.

Log-likelihood (LL) [3] is a *unithood-based* measure. For bigram terms, $ct = N_1N_2$, LL compares the observed frequency counts with the counts that would be expected, if N_1 and N_2 were co-occurring assuming independence: $P(N_1, N_2) = P(N_1)P(N_2)$. A high LL means that observed and expected values diverge significantly, indicating that N_1 and N_2 do not co-occur by chance. Contrarily, a LL close to 0 indicates that N_1 and N_2 co-occur by chance.

For the computation, two tables are created. The first one, OT, holds the observed counts taken from the corpus. The second, ET, contains the expected values assuming independence (table 1). LL can then be calculated using equation 10, where n_{ij} is the i, j cell of OT, m_{ij} is the i, j cell of ET and $T = \sum_i^j n_{ij}$.

$$LL = 2 * \sum_{i,j} n_{ij} \cdot \log\left(\frac{n_{ij}}{m_{ij}}\right), \quad \text{where} \quad m_{ij} = \frac{\sum_k n_{ik} * \sum_k n_{kj}}{T} \tag{10}$$

For N-grams, where $N > 2$, there are more than one hypothesized models to compare against the observed counts. For example, table 2 shows the different hypothesized models for trigrams. We use the extended LL [11], in order to

Table 1. Observed (OT) and expected (ET) value tables. Bigram: "gene expression".

OT	N_1	$\neg N_1$
N_2	$n_{11} = 563$	$n_{12} = 702$
$\neg N_2$	$n_{21} = 1,085$	$n_{22} = 57,553$

ET	N_1	$\neg N_1$
N_2	$m_{11} = 35.44$	$m_{12} = 1,229.56$
$\neg N_2$	$m_{21} = 1,612.56$	$m_{22} = 55,940.44$

Table 2. Hypothesized models for trigrams

$\text{Model}_1 = \frac{P(N_1N_2N_3)}{(P(N_1)P(N_2)P(N_3))}$	$\text{Model}_2 = \frac{P(N_1N_2N_3)}{(P(N_1N_2)P(N_3))}$
$\text{Model}_3 = \frac{P(N_1N_2N_3)}{(P(N_1)P(N_2N_3))}$	$\text{Model}_4 = \frac{P(N_1N_2N_3)}{(P(N_1N_3)P(N_2))}$

calculate LL values for each hypothesized model. For each model a different table of expected values is computed, while the observed values table remains the same for all. Then, for each model LL is calculated (equation 10). The model with the lowest LL value best represents the N-gram, since when a model is a good fit the observed values are close to the expected ones.

Pointwise mutual information (PMI) [1] is an information theoretic measure applied for N-gram terms. For bigrams, PMI quantifies the distance between the joint distribution of N_1 and N_2 and the joint distribution if N_1 and N_2 were independent. Equation 11 shows the PMI formula for bigram terms. If N_1, N_2 are independent: $P(N_1, N_2) = P(N_1) * P(N_2)$, then PMI is 0. For N-grams of $N > 2$, there are more than one hypothesized models to compare against the joint distribution of N-gram constituents. The process is similar to the process followed in LL. For each model we calculate different PMI values, and we choose the one with the lowest PMI value, i.e. the model which best represents the observed counts. For example, the PMI formula for the i^{th} 3-gram model of table 2 is $log(\text{Model}_i)$.

$$PMI(N_1, N_2) = \log \frac{P(N_1, N_2)}{P(N_1)p(N_2)} \tag{11}$$

3 Evaluation

3.1 Experimental Setting

For evaluation, the *GENIA* [6] and the *PennBioIE* [9] were used (table 3). Both corpora consist of *MEDLINE* abstracts, 2,000 and 2,257 respectively, and their terms are manually annotated.

For *PennBioIE* [9] evaluation we excluded annotations of quantitative values and units. In *GENIA*, annotation terms are not part of the text, but of separate *xml* attributes. Thus, *GENIA* gold standard (GS) is created by collecting these *xml* values and cleaning most non-alphanumerical characters. We observed that

Table 3. *GENIA* and *PennBioIE* corpus statistics

	sentences	tokens	terms	distinct terms	terms types
GENIA	18,546	454,848	97,876	35,947	36
PennBioIE	32,692	712,551	76,535	13,759	22

Table 4. *GS* term counts and candidate term counts per ling. filter and term length.

	GENIA					*PennBioIE*				
Length	GS	N	A&N	J&K	J&K#	GS	N	A&N	J&K	J&K#
Any	28,142	29,751	69,457	85,978	138,251	7,447	46,519	80,205	99,194	178,939
2-grams	12,654	17,103	33,021	33,021	36,866	4,034	28,489	44,072	44,072	58,086
3-grams	9,051	8,813	21,401	28,071	37,146	1,820	11,421	22,530	31,930	49,570
4-grams	3,839	3,199	9,356	15,204	29,803	821	4,157	8,629	14,945	35,746
5-grams	1,559	1,020	3,699	6,339	18,099	388	1,486	3,070	5,447	20,019
6-grams	606	297	1,317	2,239	9,005	207	694	1,172	1,822	9,105

in a few cases annotation tokens are not lemmatized (e.g. "activators of transcription", "activating function") or erroneous (e.g. "latent proviru"). However, we hypothesize that a corpus with low level of noise is acceptable for our purposes. Both *GENIA* and *PennBioIE* text was similarly cleaned. Then, both corpora were tokenized and part-of-speech (*PoS*) tagged using the GENIA tagger[2].

The first and sixth column of table 4 shows *GS* term counts of *GENIA* and *PennBioIE*, respectively. The following columns present candidate term counts, identified by each linguistic filter, for each corpus. The filters are shown in order of descending strictness. For example, the *A&N* filter identified far fewer candidates than the *J&K*. However, even the most strict filter, *Nouns*, creates more candidate terms than the valid ones. Note that, for each column, the count of candidates of any length (row 1, table 4) is not equal to the sum of all N-grams, because candidates of any length include sequences up to 12 tokens long.

The standard evaluation metrics Precision (P) and Recall (R) [12,15] (equation 12) were used for evaluating *ATR* statistical methods. *F-Score* is defined as the weighted harmonic mean of P and R: $2\left(R^{-1}+P^{-1}\right)^{-1}$.

$$P = \frac{\#\text{ correctly identified terms}}{\#\text{ identified terms}} \qquad R = \frac{\#\text{ correctly identified terms}}{\#\ GS\text{ terms}} \tag{12}$$

Table 5 shows R and P for every linguistic filter for candidates of any length and N-grams for both corpora. We observe that the less strict a filter is, the higher the R and the lower the P. *A&N* seems to achieve the best compromise between R and P. *ATR* statistical methods re-rank the list of candidates, with a target to output the actual terms higher. Thus, considering the whole list, the performance of all statistical methods is the same (table 5).

[2] www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger

Table 5. R (%) and P (%) per linguistic filter and length of candidate term

	GENIA								*PennBioIE*							
	Nouns		A&N		J&K		J&K#		Nouns		A&N		J&K		J&K#	
Length	*R*	*P*	*R*	*P*	*R*	*P*	*R*	*P*	*R*	*P*	*R*	*P*	*R*	*P*	*R*	*P*
Any	35.4	33.5	80.2	32.5	80.2	26.3	85.4	17.4	37.2	6.0	63.1	5.9	63.7	4.8	76.1	3.2
2-grams	48.1	35.6	88.0	33.7	88.0	33.7	90.6	31.1	52.6	7.5	78.1	7.1	78.0	7.1	90.6	6.3
3-grams	31.9	32.8	80.4	34.0	80.5	25.9	84.5	20.6	26.8	4.3	60.7	4.9	61.6	3.5	73.1	2.7
4-grams	21.3	25.5	67.0	28.7	70.4	17.8	78.9	10.2	15.8	3.1	42.8	4.1	45.2	2.5	56.5	1.3
5-grams	14.9	22.7	63.8	26.9	64.2	15.8	77.0	6.6	4.7	1.2	17.1	2.2	18.7	1.3	38.3	0.7
6-grams	9.2	18.9	54.5	25.1	54.5	14.7	71.0	4.8	3.9	1.3	13.0	2.3	13.5	1.5	24.6	0.6

Table 6. Executed experiments on each corpus

Candidate term length	Any, 2-grams, 3-grams, 4-grams, 5-grams, 6-grams
Linguistic filter	Nouns, A&N, J&K, J&K#
ATR stat. approach	*NC Value*, *PMI* (*N*-grams only) *LL* (*N*-grams only), *SB* (Nouns and A&J only)

As discussed in section 2, the *Log-likelihood* (*LL*) method can only be applied separately for sequences of a specific length. We implemented the extended *LL* algorithm for N-grams, $N \in [2, 6]$. There are only 433 *GENIA GS* terms and 177 *PennBioIE GS* terms longer than 6 tokens, very few to experiment with (table 4). The results of the *LL* algorithm for different values of N are not comparable to each other. Thus, we set separate experiments up for each value of $N \in [2, 6]$.

For example, for 2-grams we first apply a linguistic filter to identify candidates of which we keep 2-grams only. Next, 2-grams are re-ranked according to one of the implemented statistical methods. Evaluation is performed towards the 2-gram *GS* terms. Experiments for the other values of N were set up identically.

Except for N-grams, we ran experiments taking into account sequences of any length, higher than 2. For each one, candidate terms are identified using one of our four linguistic filters. Then, one of *C-Value*, *NC-Value* or *SB* re-ranking method is applied. Evaluation uses the whole *GS* term set. Note that the *SB* method makes sense only when following the *Nouns* or the *A&N* linguistic filter.

The *NC Value* algorithm takes as input a list of candidates, ranked by the *C Value* algorithm and is subject to two parameters: the percentage of the list, starting from the top, that it will take into account to identify context terms and the size of the context window. We experimented using values 5%, 7.5% and 10% for the former one and 2, 4, 6, 8, 10 for the latter.

Table 6 shows all executed experiments, referring to the combination of length of candidate terms, filtering and statistical approach used. To visualise the results, we used an approach similar to the one indicated in [16]. R and P values were calculated at 0.5% increments on the list of candidates and plotted on graphs, such as figure 2. For each increment on the list, P refers to the ratio of true positives over the overall number of candidates and R refers to the ratio of

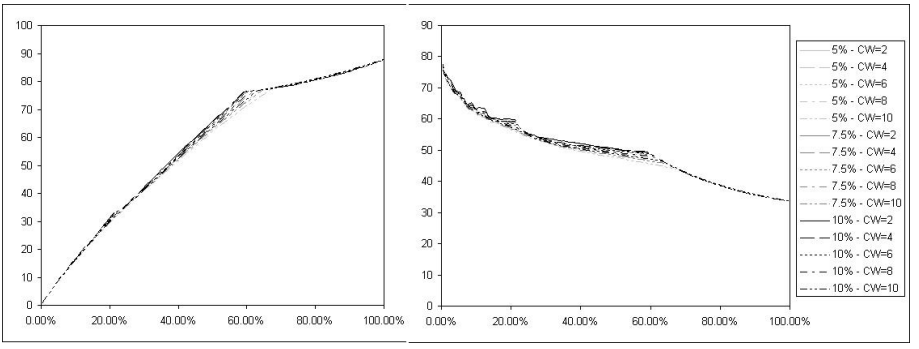

Fig. 2. *GENIA* 2-grams, J&K filter, *NC Value* results, *R* and *P*

true positives over the number of *GS* terms. The x-axis shows the percentage of the list taken into account. *Frequency of occurrence* (*FR*) is used as baseline.

Intuitively, the *P* curve of a bad performing method would be relatively horizontal indicating that the true positives were dispersed uniformly throughout the list rather than pushed towards the top. Contrarily, the *P* curve of a well-performing method would be 100% until the percentage point at which all *GS* terms would have been retrieved, where a sharp decrease would occur [11].

3.2 Results

Figure 2 shows the 2-gram *P* and *R* curves of *NC Value* for 15 parameter combinations (see subsection 3.1), using the *J&K* linguistic filter on *GENIA* corpus. We observe that different combinations do not affect the results. This behaviour remains the same for all linguistic filters and for all term lengths. Interestingly, for all the above experiments the performance of *C* and *NC Value* is almost identical, both for *GENIA* and *PennBioIE*.

Figure 3 shows the *F-Score* performance for 3-gram candidate terms of *GENIA* and *PennBioIE* as identified by the *Nouns* linguistic filter. We observe that *termhood-based* methods outperform *unithood-based* ones. *SB*, *C* and *NC Value* perform similarly with *SB* having a slightly better *F-Score* on *GENIA*. *PMI* curves are below the baseline on both corpora. On the contrary, *LL* outperforms the baseline of *FR* on *PennBioIE* but not on *GENIA*. Possible reasons for the behaviour of *LL* and *PMI* are discussed in subsection 3.3. The ranking of *ATR* methods remains the same as in figure 3 for any *N*-gram using both the *Nouns* and the *A&N* linguistic filter, on both corpora.

The performance for *N*-gram candidate terms as identified by *J&K* and *J&K#* demonstrate the following trends: On *GENIA* the highest performance is achieved by *C* and *NC Value* methods throughout the plots. The remaining methods in order of decreasing *F-Score* are: *FR*, *LL* and *PMI*. The bigger *N* is, the closer *FR*, *LL* and *PMI* curves are to each other.

On *PennBioIE*, the performance differences between *FR*, *LL*, *C* and *NC Value* are insignificant, while *PMI* clearly performs worse. In this corpus we observe

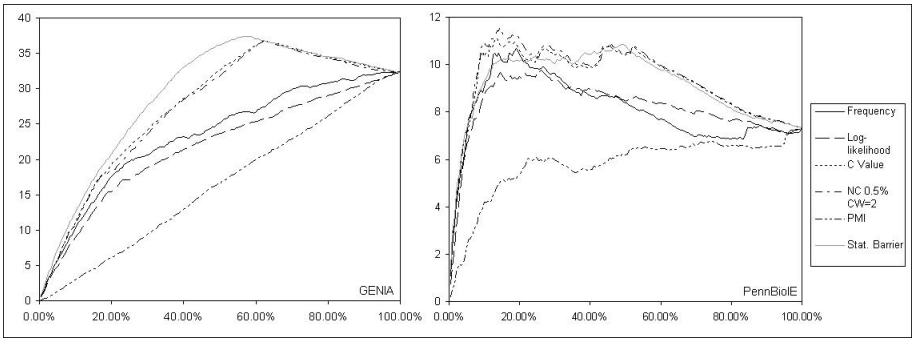

Fig. 3. *GENIA* and *PennBioIE* 3-grams, Noun filter, *F-Score*

that *termhood-based* methods have a comparable performance with the baseline. 6-gram results follow the same trends in general, but they are not very reliable due to the small number of candidates.

On both corpora for candidates of any length identified by *Nouns* and *A&N*, *SB*, *C* and *NC Value* methods exceed the baseline of *FR*, achieving similar levels of performance. Using the *J&K* and *J&K#* on *GENIA* (*PennBioIE*), the performances of *C*, *NC Value* and *FR* are similar for increments up to 10% (on both corpora) of the candidate list. For increments between 10% and 30% (50% for *PennBioIE*), *FR* performs better than *C* and *NC Value*. After 30% (50%), *C* and *NC Value* perform better than *FR*.

3.3 Discussion

Our results (section 3.2) show that *termhood-based* methods re-rank the candidate list better than *unithood-based* methods or equally well, irrespective of the candidate terms length and linguistic filter used. A possible reason is that *unithood-based methods* measure the strength of attachment of the candidate term constituents, in effect assigning high scores to candidate terms, which might not refer to domain concepts. For example, in *GENIA*, *"allergic inflammatory"*, substring of the term *"allergic inflammatory disease"*, occurs at least equally often as the term, although the former is not a term itself.

The only setting in which a *unithood-based* method (*LL*) performed equally well to the *termhood-based* methods was when using *J&K* or *J&K#* to extract *N*-gram candidates from *PennBioIE*. A possible explanation for this peculiarity is the limited amount of nestedness information in *PennBioIE*, which degrades the performance of *termhood-based* approaches. Particularly for 3-grams, the average nested frequency in *PennBioIE* is 1.03, while in *GENIA* is 1.16. Note that *PennBioIE* is almost double the size of *GENIA* (table 3).

PMI overestimates rare events, which dominate the candidate term lists. For example, *A&N* identifies 69,457 *GENIA* candidate terms, out of which 52,998 (76.3%) occur only once, and 16,459 twice. *LL* outperforms *PMI*, due to its milder tendency in overestimating rare events.

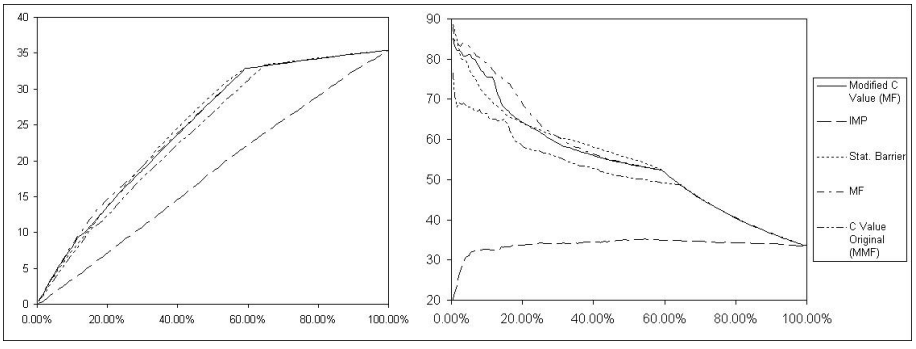

Fig. 4. *GENIA* sequences of any length, *Nouns* filter, various methods, R and P

C and *NC Value* exploit nestedness information, in the sense that the more often a candidate appears as nested, the less likely it is a valid term. *SB* considers this information through *MF* counts. *NC Value* attempts to improve *C Value* by exploiting contextual information. However, unsuccessfully, under our evaluation scheme. To investigate this, we adjusted the interpolation constant of equation 7 to assess the contribution of the CF only ($NCV(ct) = 0 * CV(ct) + 1 * CF(ct)$). P curves are almost uniform across most of the plot.

SB exploits two sources of information: Firstly, *IMP* (equation 8), assumes that complex terms consist of existing simple terms. Secondly, *MF* (equation 9), refers to the marginal frequency counts. To evaluate the contribution of each, we executed two experiments, which re-rank the candidate term list taking into account *IMP* and *MF* separately. Interestingly, P of *IMP* is roughly uniform on *GENIA* (figure 4), which means that it contributes negatively to *SB*. On the contrary, *MF* successfully redistributes candidates towards the top of the list. Thus, the corresponding P curve is higher than the curve of *SB* in the x-axis interval [0%, 30%]. *PennBioIE* experiments verified these results.

C Value suggests that the higher the nested frequency of a candidate term, ct, the less likely it is a valid term, conditional to the number of distinct candidate terms, in which ct appears as nested. Hence, *C Value* calculates a weighted version of marginal frequency (*MMF*), $f(ct) - NST(ct)$ (formula 5). $NST(ct)$ is the ratio of the frequency of the candidate as nested over the number of distinct terms, in which it appears nested. To examine the effect of *MMF* in *C Value*, we replaced the *MMF* in the *C Value* formula with *MF*. Results show that the modified version of *C Value* performs better i.e. *MF* captures nestedness better than *MMF*. However, *MF* outperforms even this modified version of *C Value*, for increments up to 25% of the candidate list for *GENIA* and 55% for *PennBioIE*.

4 Conclusion

We reviewed and evaluated state-of-the-art linguistic filtering and statistical *ATR* methods under a common evaluation scheme. Our results indicate that:

(1) *termhood-based* methods have in general superior performance over *unithood-based* ones, and (2) that the number of independent occurrences of a candidate term is the most effective source of nestedness information for *ATR*.

References

1. Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. Computational Linguistics 16(1), 22–29 (1990)
2. Dias, G., Kaalep, H., Muischnek, K.: Automatic Extraction of Verb Phrases from Annotated Corpora: A Linguistic Evaluation for Estonian. In: EACL/ACL Workshop on Collocations, Toulouse, France (2001)
3. Dunning, T.E.: Accurate Methods for the Statistics of Surprise and Coincidence. Computational Linguistics 19(1), 61–74 (1993)
4. Evert, S., Krenn, B.: Methods for the qualitative evaluation of lexical association measures. In: ACL, Morristown, NJ, USA (2001)
5. Frantzi, K.T., Ananiadou, S., Mima, H.: Automatic recognition of multi-word terms: the C-value/NC-value method. International Journal on Digital Libraries 3(2), 115–130 (2000)
6. Gu, B.: Recognizing Nested Named Entities in GENIA corpus. In: HLT-NAACL BioNLP Workshop, New York, pp. 112–113 (2006)
7. Justeson, J.S., Katz, S.M.: Technical terminology: some linguistic properties and an algorithm for identification in text. Natural Language Engineering 1(1), 9–27 (1995)
8. Kageura, K., Umino, B.: Methods of automatic term recognition: a review. Terminology 3(2), 259–289 (1996)
9. Kulick, S., Bies, A., Liberman, M., Mandel, M., Mcdonald, R., Palmer, M., Schein, A., Ungar, L., Winters, S., White, P.: Integrated Annotation for Biomedical Information Extraction. In: Hirschman, L., Pustejovsky, J. (eds.) HLT-NAACL BioLINK Workshop, Boston, Massachusetts, USA, pp. 61–68 (2004)
10. Manning, C., Schutze, H.: Foundations of Statistical Natural Language Processing. Chapter: Collocations. MIT Press, Cambridge (1999)
11. Mcinnes, B.T.: Extending the Log Likelihood Measure to Improve Collocation Identification. Master's thesis. University of Minnesota (2004)
12. Mikheev, A., Moens, M., Grover, C.: Named Entity recognition without gazetteers. In: EACL, Bergen, Norway, pp. 1–8 (1999)
13. Nakagawa, H.: Automatic Term Recognition based on Statistics of Compound Nouns. Terminology 6(2), 195–210 (2000)
14. Pecina, P., Schlesinger, P.: Combining Association Measures for Collocation Extraction. In: ACL, Sydney, Australia (2006)
15. Radev, D., Teufel, S., Saggion, H., Lam, W., Blitzer, J., Qi, H., Elebi, A., Liu, D., Drabek, E.: Evaluation challenges in large-scale document summarization. In: ACL, Sapporo, Japan (2003)
16. Wermter, J., Hahn, U.: Collocation extraction based on modifiability statistics. In: COLING, Morristown, NJ, USA (2004)