AUTOMATIC TERMINOLOGY EXTRACTION FOR THEMATIC CORPUS BASED ON SUBTERM CO-OCCURRENCE

Chunyu Kit

Computational Linguistics Program
Philosophy Department
Carnegie Mellon University
Pittsburgh, PA 15213, USA
cykit@lcl.cmu.edu

Abstract

In this paper, we present a multi-word terminology extractor for thematic corpus based upon the co-occurrence of subterms. With regard to the basic properties of terminologies, among which we emphasize the structural dependency relation between subterms, a number of straightforward hypotheses are proposed as strategies for terminology recognition. The key idea to measure the structural dependency within a corpus-based approach is that higher frequency of subterm co-occurrence may indicate higher structural dependecy. The experimental results show that our algorithm can extract multi-word terminologies with nice correspondence to domain-specific concepts and notions.

1. Introduction

We developed a practical system[1] to identify multi-word basic text units (BTUs) within a corpus based approach (Kit 1994), aiming at term space reduction for a phrase-based IR system like CLARIT (Evans et al. 1991b, 1993a, 1993b; Paijmans, 1992). It is observed that BTUs are a small subset of raw NPs[2] that are more conceptually important. Most BTUs recognized are concept[3]-like collocations or compounds of words and domain-specific terminologies. Experimental results on several standard IR corpora showed that we can use BTUs to substitute for full terms (i.e., all raw NPs), since using BTUs has equivalent IR performance as using full terms but leads to about 50% term space reduction (Kit 1994). This positive effect would be especially valuable for large-scale IR tasks, because it promises a great efficiency enhancement in phrase-based IR.

Another target of the research is sublexicon discovery for thematic corpus. We found that the BTU recognition techniques are applicable to multi-word terminology extraction. We notice that multi-word terminologies are a subset of BTUs which have more restricted correspondence to domain-specific concepts and notions. This is the starting point, and also the basis, for us to modify the BTU recognition algorithm for multi-word terminology extraction.

In literature, several researchers reported their analytical approaches to terminology recognition, for example, Ananiadou (1988), Bourigault (1992), with focus upon the syntactic structure analysis. Other related research can be found scattered in studies of terminology processing (Sager 1990), noun-noun compounds (Levi 1978; Rackow et al. 1992), tokenization of words in Asian languages like Chinese and Japanese (Liang 1984; Chen and Liu 1992; Webster and Kit 1992), recognition of idioms and collocations of words for MT and other NLP tasks (Smadjia 1990, 1991; Kit and Webster 1992), etc. In this paper, we present a corpus based approach to terminology extraction with statistical structure analysis. One of its distinctive features lies in that it makes use of the co-occurrence frequency of subterms[4] as a measure for the structural dependency relation between subterms in determining whether a phrase can be recognized as a multi-word terminology.

[1]This research was supported by the CLARIT Project, Laboratory for Computational Linguistics, Carnegie Mellon University and the Corporation for National Research Initiatives/ARPA "Computer Science Technical Reports" Project. CLARITECH Corporation provided many facilities for the experiments. The comments, advice, support, help and encouragement from David A. Evans, Bob Carpenter, Teddy Seidenfeld, Lori Levin, Nataša Milić, Robert G. Lefferts, Gregory T. Grefenstette, Yeyi Wang, Xiang Tong and Connie Bartusis are gratefully acknowledged.

[2]As reported in Kit (1994), only 20-30% raw NPs are recognized as BTUs.

[3]The term *concept* is used in an intuitive sense in many cases. In relation to terminology extraction, it can be understood in a more restricted *terminological* sense.

[4]A word or a shorter term nested in a longer term or phrase as a constituent is referred to as a *subterm*.

The methodology adopted here includes (1) selected NLP techniques, in particular, the CLARIT NP parser to recognize raw NPs without giving precise structure analysis; (2) statistics, e.g., the co-occurrence frequency of subterms in longer phrases; (3) heuristics of combining the above two to achieve our goal of terminology extraction, e.g., we deal with 2-word phrases first, then 3-word/subterm phrases, and so on. Experimental results show that multi-word terms extracted within this approach have nice correspondence to domain-specific concepts and notions.

In the following sections, we will first discuss the strategies based on which we develop an algorithm for discovery of multi-word BTUs, and then the modification of this algorithm for terminology extraction. An experiment using this algorithm on a linguistic corpus is also reported.

2. Strategies for Automatic Discovery of BTUs

2.1 Basic Properties of BTUs

In order to develop appropriate strategies to discover BTUs, we must first have a good understanding of their properties. Based on previous studies on compounds and terminology, we emphasize the following basic properties of multi-word BTUs:

Syntactic persistency: Constituents or subterms in a multi-word BTU usually hold a rather stable syntactic relation one another. Such syntactic connection is not broken under normal condition. Hypothesis 2 below is proposed to respect this basic property.

Productive combination: BTUs are productive, in the sense that they combine with other words or BTUs to yield many new phrases. Co-occurrences of a BTU with other words or BTUs within NPs in a corpus will be an important measure on this property.

Unit-semantic denotation: A BTU bears unit-semantic content, e.g., a basic concept, a domain-specific notions, a proper name, etc., in contrast to some very general phrases like "next one", "following example", etc. We expect that BTUs to be found could bear conceptual information content.

2.2 Strategies for Discovering BTUs

We assume that the algorithm of discovering multi-word BTUs follows from the hypotheses proposed below with respect to the above properties of BTUs as well as to our intuition and common sense.

Hypothesis 1: A BTU, or a term,[5] can be used independently.

That is, a BTU must exist independently (i.e., as itself being an NP) somewhere in the corpus. For example, "previous chest" from the phrase "previous chest examination" and "active lung" from "active lung disease" are not likely to be terms in a medical corpus.

Hypothesis 2: The structural dependency connection between constituents (a single-word or multi-word subterm) in a BTU cannot be broken throughout the whole corpus.

That is, for a term $< A\ B >$, in which A and B are subterms, there should not be any instance of a word sequence in the form $< A\ C\ B >$ such that the structural dependency relation between A and B is broken by C. For example, "hot dog" is a kind of food, whereas "hot $\cdots$ dog" (if any) would probably be a kind of animal and it is unlikely to be a BTU.

Note, however, that if the dependency connection is interpreted as *continuously co-occurring*, it will lead, unexpectedly, to a too strong hypothesis that rules out many potential BTUs. Consider, for example, the following phrases, most of which could be terms in a medical corpus:

active disease $\Longrightarrow$ *active* lung *disease*
active infectious *disease*
active contagious *disease*
......

Although there are so many instances in which "active" and "disease" are separated from one another, we cannot deny that "active disease" is indeed a term. We can find that in all longer phrases like the above ones, the dependency relation between "active" and "disease" remain the same: the former is a modifier and the latter is the head. Such dependency relation appears to be ***structurally definable***, e.g., modifiers and complements are dependent upon their heads, no matter the two words/subterms are continuously or discontinuously co-occurring.

More importantly, it is reasonable to assume, within a corpus based approach, that this kind of structural dependency relation is ***statistically recognizable***, in particular, in the case of structural ambiguity. In general, a parser is able to assign structure to a phrase, but in the ambiguous cases, for example, whether "active" modifies "lung" or "disease" in the phrase "active lung disease", we still need to resolve the structural ambiguity by

[5]The terms *basic textual unit* (BTU), *term* and *terminology* are used interexchangeably in some cases in this paper. A *term* can be understood as a terminology or as a term for text indexing in IR, depending on the context.

some appropriate statistical means.

2.3 Statistical Dependency vs. Structure Analysis

To resolve the structural ambiguities of this type, we propose a simple statistical approach, in contrast to a syntactic structure analysis. Pure structure analysis appears to be too expensive for computation, in the sense that it needs a sophisticated parsing process, and to have low effect upon structural disambiguation, e.g., it is unable to resolve the ambiguity in all phrases with $(A\ N\ N)$ pattern, like "active lung disease". The statistical approach is proposed as simple as the following: higher frequency of co-occurrences of two words or subterms within a specific structural (syntactic) category, e.g., NP, indicates a higher structural dependency connection in that category.

For example, in order to determine whether "active disease" can be a term, we need to examine the structural dependency relations between these two words in the phrase "active lung disease". The corresponding attribute list $(A\ N\ N)$ of this phrase leads to the following structural ambiguity:

$$(a)\ ((A\ N)\ N) \qquad (b)\ (A\ (N\ N))$$

If case (a), where "active" and "lung" are treated as having higher structural dependency each other, is confirmed by statistical data of co-occurrence frequency, i.e.,

$$freq(active,\ lung)) > freq(active,\ disease)$$

then "active disease" will be statistically ruled out as a term.[6] Otherwise, we assume (b), where "active" is treated as modifier to "disease" rather than to "lung". This is not a negative evidence against "active disease" being a term, with respect to the Hypothesis 2. So, in order to determine whether "active disease" is a term, it is necessary to examine all similar discontinuous co-occurrences of "active" and "disease" in the corpus in question.

2.4 Inadequacy of Some Statistical Measures on Structural Dependency

Structural ambiguities take place so often in multi-word NPs. Any multi-word NPs containing two or more nouns can be structurally ambiguous. We need some kind of statistical measure to determine the structural dependency relation between words/subterms in an ambiguous case.

[6]More details on measuring the structural dependency between subterms follows in next sections.

The original measure for the dependency of two events is given in Bayesan statistics as conditional probability:

$$P(x|y) = \frac{P(xy)}{P(y)}$$

Church (1989, 1991) uses *mutual information* to describe the word association relation between two words, which appears to derive from Bayesan statistics. It is formulated as below:

$$I(x;y) = log_2 \frac{P(x,y)}{P(x)\ P(y)}$$

Wilks et al. (1990) define a number of *relatedness functions* as statistical measures to describe the relatedness of words based on their co-occurrence in a corpus. One of these functions is *dependency extraction* between two words, formulated as the following:

$$dex(x,y) = \frac{f_{xy} - f_x \cdot f_y}{min(f_x, f_y) - f_x \cdot f_y}$$

However, according to our observation, measures of these kinds are not appropriate to describe structural dependency. Let us take "hot dog" as an example to show this. We may have a corpus, for example, the conversations of daily life, in which both individual words "hot" and "dog" have a frequency much higher than the frequency of the collocation "hot dog", i.e.,

$$freq(hot) \gg freq(hot\ dog)$$
$$freq(dog) \gg freq(hot\ dog)$$

This has the result that $P(hot|dog)$, $P(dog|hot)$, $I(hot;\ dog)$ and $dex(hot, dog)$ are all very small such that none of them is significant enough to indicate the real structural dependency between "hot" and "dog" in such case. Rather, it may give the misleading result that "hot" and "dog" have a very loose structural relation in the collocation "hot dog", contradicting the fact that their structural dependency is rather high.

It can be observed that such misleading measures resulted from the fact that a great number of irrelevant "hot"s and "dog"s, which occur independently of one another, are counted as the statistical factors to decide the structural dependency relation between the specific "hot" and "dog" in "hot dog".

2.5 Subterm Co-occurrence Frequency: A Measure for Structural Dependency

In order to measure the structural dependency between two words/subterms in a more reliable way, we need to eliminate as much as possible the independent occurrences of each word in question.

Since the independent occurrences of one word tell nothing about the structural dependency between itself and the other, both can be viewed as "noise" data upon the structural dependency relation between the two words. We need to prevent such noisy data from obstructing the measure of structural dependency. For this purpose, we propose the following hypothesis:

Hypothesis 3: Higher co-occurrences frequency within a specific structural category may indicate higher structural dependency.

There are three different cases in this hypothesis, as stated in the following strategies:

Strategy 3.1: Higher frequency of co-occurrences (*fco* hereafter) of two words/sub-terms within a specific structural category, e.g., NP, may indicate higher structural dependency.

Strategy 3.2: Higher frequency of continuous co-occurrences (*fcco* hereafter) of two words/sub-terms within a specific structural category, e.g., NP, may indicate higher structural dependency.

Strategy 3.3: Higher frequency of independent continuous co-occurrences (*ficco* hereafter) of two words/sub-terms as a specific structural category, e.g., NP, may indicate higher structural dependency.

However, a problem with these strategies is how much a difference of co-occurrence frequency is significant enough to indicate the difference in structural dependency (*sdep* hereafter)? If we have the measures like the following, for example,

$$fco(w_1, w_2) = 588$$
$$fco(w_1, w_3) = 583$$

we are not sure whether these are adequate to predict that

$$sdep(w_1, w_2) > sdep(w_1, w_3).$$

There should be a significance factor to resolve this problem, for example, a factor of 2 or 3 times, which means that only if

$$fco(w_1, w_2) > 2 \cdot fco(w_1, w_3)$$

can we then say

$$sdep(w_1, w_2) > sdep(w_1, w_3).$$

The significance factors for fco, $fcco$ and $ficco$ may be different. Appropriate factors for comparing fco, $fcco$ and $ficco$ should be obtained from experiments or expert experience.[7]

The relation between these three strategies are the following: Strategy 3.2 will be applied if the difference of *fco*'s (in Strategy 3.1) is not significant enough to tell the difference of *sdep*'s; Strategy 3.3 will be applied if $fcco$ difference (in Strategy 3.2) is not significant. In a case that all three of the above types of measures are not significant enough to indicate the structural dependency preference, we assume that it has no effect on the determination of whether a sequence of words is a term.

[7]The significance factors for fco, $fcco$ and $ficco$ used in our experiments reported below are 1.5, 3 and 5, respectively.

3. A Terminology Extraction Experiment

Following the above strategies, we implemented a BTU recognition system for phrase-based IR. It is further modified into a terminology extractor for thematic corpus. The main modification is to add a stop-list[8] to filter out the non-terminological phrases which have high frequency but are too general to be terminologies, e.g., "next one", "same way", etc.

In order to examine the unit-semantic denotation of the extracted terms, that is, how well they correspond to domain-specific concepts and notions, we conducted terminology extraction experiments on several corpora. The one reported here is on Bob Carpenter's manuscript *Lecture Notes on Natural Language Semantics.*[9] Here is the general information about the corpus and the terminology extraction:

·Size: 523 Kilobytes[10]
·Number of words: 87K
·Number of unique multi-word raw NPs: 4.4K
·Number of unique single word NPs: 1.5K
·Number of unique words in all NPs: 2.8K
·Number of extracted terms: 0.8K

With the aid of a stop-list, about 800 multiword terms (about 18% of raw NPs) are extracted as multi-word terminologies. Some sample fragments of the extracted terms in the high, medium and low frequency areas are given in Appendix A.

Note that the inconsistent information on the numbers of words and syntactic categories is produced by the NP parser, for example, the two-word phrases "modal logic" and "phrase structure" are each attached with only one syntactic category. This reveals that they are treated as compounds like a word in the NP parser's lexicon. Regardless of such inconsistent information,

[8]It is a traditional method in IR. For example, *the, a, you, my, they,* etc., are typical stop-list words.

[9]Bob Carpenter. 1993. *Lecture Notes on Natural Language Semantics.* ms. Computational Linguistics Program, Philosophy Department, Carnegie Mellon University. It is currently in press by MIT Press in the title of *Type Theoretical Semantics.*

[10]Exclusive of Latex formats, formula and pictures.

the terminology extractor also recognizes them as terms with the aid of statistical data. This illustrates, partially though, that the extractor works in a right way.

However, the terminology extractor is purely based upon statistical data on co-occurrence of subterms and makes use of little knowledge or semantic information. It is inevitable that it is fooled by some high frequency non-terminological phrases like "following example", "following sentence", etc., in the corpus. In order to get rid of such noisy information on terminology extraction, it is necessary to have a stop-list as a filter. A fragment of the stop-list added to the terminology extractor looks like the following:

```
following      consisting     resulting
interesting    thing          being
beginning      adding         deriving
updating       defining       example
sample         very           across
the            drop           particular
previous       serious        step
component      kind           whole
entire         instance       important
importance     perspective    simple
simplest       present        one ......
```

A stop-list word like "following", for example, filters out non-terminological phrases as those given in Appendix B, some of which are of very high frequency.

The extracted terms are evaluated by the first and second year graduate students in the Computational Linguistics Program at CMU, who used or are using the manuscript as text book in the semantic class. The following is the statistics of the overall evaluation on "how well the recognized terms correspond to domain-specific concepts and notions":

```
0 - excellent;
7 - better than good;
1 - good;
1 - just OK;
0 - less than OK, i.e., bad;
0 - very bad.
```

Most evaluators choose "better than good" as overall evaluation among the 6 choices. The author of the manuscript also confirms that "most extracted phrases look like terms", in addition to having pointed out some bad terms. About 50 phrases, i.e., 6% of the extracted terms, are pointed out by evaluators to be bad terms.

The corpus used is a small one,[11] and the experimental result turns out to be satisfiably good. Since the terminology extractor relies heavily upon statistical data, we have reason to believe it to have better performance if working on larger corpora.

[11] We chose this small corpus to report here only for the sake of the appropriate evaluation available from those who are familiar with it.

4. Conclusion

This is a preliminary study on terminology extraction using the BTU recognition algorithm we have developed. There are many things to be improved, for example, how to select better stop-list words for a thematic corpus, how to use domain knowledge, etc.

However, through the terminology extraction experiments, we can see that most extracted terms have nice correspondence to domain-specific concepts and notions. This can be a piece of evidence for that the BTUs and terminologies recognized by the algorithm are conceptually important. They have nice unit-semantic denotation, i.e., they are concept-like information units. Therefore, we believe that the terminology extractor can be a useful tool for practical terminology processing, for example, automatic construction of term banks, discovery of domain-specific sublexicon, etc.

4.1 A Word About Single-Word Terms

At first sight, it is really unlikely for a computer without expert knowledge to determine whether a single word can be a terminology in a domain. Our work reported above focuses only upon multi-word terms, however, the result is believed to be helpful to recognize single-word terms. Intuitively, we may propose the following:

Hypothesis 4: An independent single word with higher occurrence frequency in multi-word terms is more likely to be a single-word term.

To an extent, this hypothesis can distinguish stop-list words from words with concrete semantic content, since multi-word terms contain few stop-list words. So, it can inherently prevent stop-list words from getting into single-word terms. This could be a starting point to develop a more sophisticated strategy to incorporate single-word terminology recognition into our algorithm, with the aid of other resources.

5. References

[1] Ananiadou, S. 1988. *Towards a Methodology for Automatic Term Recognition*, PhD dissertation, University of Manchester.

[2] Bourigault, Didier. 1992. "Surface Grammar Analysis for the Extraction of Terminological Noun Phrases", in *COLING-92*, 977-981, August 1992, Nantes, France.

[3] Chen, Keh-Jian and Shing-Huan Liu. 1990. "Word Identification for Mandarin Chinese Sentences", in

COLING-92, 101-107, Aug., 1992, Nantes, France.

[4] Church, Kenneth W. 1988. "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text", in *Proceedings of the Second Conference on Applied Natural Language Processing*, 136-143, 1988.

[5] Church, Kenneth W. 1989. "Word Association Norms, Mutual Information, and Lexicography", in *ACL-27*, 1989, Vancouver.

[6] Church, Kenneth W., William Gale, Patrik Hanks and Donald Hindle. 1991. "Using Statistics in Lexical Analysis", in Uri Zernik (ed.), *Lexical Acquisition: Exploiting On-Line Resources To Build A Lexicon*, 115-164, NJ: Lawrence Erlbaum Assoc.

[7] Evans, David A., Kimberly Ginther-Webster, Mary Hart, Robert G. Lefferts and Ira A. Monarch. 1991a. "Automatic Indexing Using Selective NLP and First-Order Thesauri", in A. Lichnerowicz (ed.), *Intelligent Text and Image Handling: Proceedings of the Conference, RIAO'91*, 624-644, Amsterdam: Elsevier.

[8] Evans, David A., Steven K. Handerson, Robert G. Lefferts and Ira A. Monarch. 1991b. *A Summary of The CLARIT Project.* Technical Report No. CMU-LCL-91-2, Laboratory for Computational Linguistics, Carnegie Mellon University, Pittsburgh, 1991, 12pp.

[9] Evans, David A., Steven K. Handerson, Ira A. Monarsh, J. Pereiro, L. Delon and W.R. Hersh. 1991c. *Mapping Vocabulary Using "Latent Semantics".* Technical Report No. CMU-LCL-91-1, Laboratory for Computational Linguistics, Carnegie Mellon University, Pittsburgh, PA, 1991, 15pp.

[10] Evans, David A., Robert G. Lefferts, Gregory Grefenstette, Steven K. Handerson, William R. Hersh and Armar A. Archbold. 1993a. "CLARIT TREC Design, Experiments, and Results". In Donna Harman (ed.), *The First Text REtrieval Conference* (TREC-1). NIST Special Publication 500-207. Washington, DC: U.S. Government Printing Office, 1993, 251-286; 494-501.

[11] Evans, David A. and Robert G. Lefferts. 1993b. "Report on the CLARIT-TREC-2 System". Laboratory for Computational Linguistics, Department of Philosophy, Carnegie Mellon University, Pittsburgh, PA.

[12] Fagan, Joel L. 1987. *Experiments in Automatic Phrase Indexing for Document Retrieval: A Comparison of Syntactic and Non-Syntactic Methods.* PhD dissertation, Department of Computer Science, Cornell University, September, 1987.

[13] Fagan, Joel L. 1989. "The Effectiveness of A Non-syntactic Approach to Automatic Phrase Indexing for Document Retrieval". *Journal of the American Society of Information Science*, 40(2):115-132, 1989.

[14] Kit, Chunyu. 1994. *Discovery of Multi-Word Basic Text Units (BTUs) in Raw NPs for Information Retrieval: A Corpus Based Approach*, Master's Project Report, Computational Linguistics Program, Philosophy Department, Carnegie Mellon University, Pittsburgh.

[15] Kit, Chunyu and Jonathan J. Webster. 1992. "Machine Translation of Idioms Based on Tokenization", in *Proc. of 1st Singapore International Conference on Intelligent Systems*, Sept., 1992, Singapore.

[16] Levi, Judith N. 1978. The Syntax and Semantics of Complex Nominals. Academic Press.

[17] Lewis, David D., and Croft, W. B. 1990. "Term Clustering of Syntactic Phrases", 385-404, *SIGIR-90.*

[18] Liang, Nanyan. 1984. "The Chinese Automatic Word Segmentation System CDWS" (in Chinese), *Journal of Beijing University of Aeronautics and Astronautics*, No.4, 1984.

[19] Paijmans, Hans. 1992. *Comparing IR Systems: CLARIT and TOPIC*, ITK Technical Report No. 39.

[20] Rackow, Ulrike, Ido Dagan and Ulrike Schwall. 1992. "Automatic Translation of Noun Compounds", in *COLING-92*, 1249-1153, August, 1992, Nantes, France.

[21] Sager, Juan C. 1990. *A Practical Course in Terminology Processing*, Amsterdam: John Benjamin.

[22] Salton, G., and McGill, M. J. 1983. *Introduction to Modern Information Retrieval.* McGraw-Hill, New York.

[23] Webster, J. and Kit, C. 1990. "Tokenization as the Initial Phase in NLP", in *COLING-92*, 1106-1110, August, 1992, Nantes, France.

[24] Wilks, Y., D. Fass, C.M. Guo, J.E. McDonald, T. Plate and M. Slator. 1990. "Providing Machine Tractable Dictionary Tools". *Machine Translation 5*, 1990, 99-154.

[25] Wilks, Y., L. Guthrie and J. Cowie. 1992. "Combining week method in large-scale text processing". In P.S. Jacobs (ed.), *Text-Based Intelligence Systems: Current Research and Practice in Information Extraction and Retrieval*, 35-58. Lawrence Erlbaum Associates.

Appendix A: Fragments of Extracted Terms[11]

sequence number		terms	syntactic categories	alone freq		total occurrences
1	=:	first order logic	ADJ NOUN NOUN	70	-->	82
2	=:	categorial grammar	ADJ NOUN	64	-->	108
3	=:	noun phrase	NOUN NOUN	55	-->	121
4	=:	higher order logic	ADJ NOUN	54	-->	54
5	=:	truth value	NOUN NOUN	52	-->	62
6	=:	lambda calculus	NOUN NOUN	43	-->	76
7	=:	modal logic	NOUN	42	-->	61
8	=:	lexical entry	ADJ NOUN	39	-->	76
9	=:	lambda term	NOUN NOUN	38	-->	54
10	=:	propositional logic	ADJ NOUN	37	-->	49
11	=:	natural language	NOUN	30	-->	68
12	=:	possible world	NOUN	28	-->	40
13	=:	simply typed lambda calculus	ADV PASTPART NOUN NOUN	26	-->	26
14	=:	beta reduction	NOUN NOUN	25	-->	32
17	=:	proof theory	NOUN	20	-->	28
18	=:	syntactic category	ADJ NOUN	20	-->	22
19	=:	natural language semantics	NOUN NOUN	19	-->	21
20	=:	introduction rule	NOUN NOUN	19	-->	34
21	=:	meaning postulate	PROG NOUN	19	-->	20
22	=:	generalized quantifier	PASTPART NOUN	19	-->	24
23	=:	eta reduction	NOUN NOUN	19	-->	25
....						
349	=:	backward application scheme	ADJ NOUN NOUN	3	-->	3
350	=:	forward application	NOUN_ADJ NOUN	3	-->	3
351	=:	pure applicative categorial grammar	ADJ ADJ ADJ NOUN	3	-->	3
352	=:	context free grammar	NOUN ADJ NOUN	3	-->	3
353	=:	arithmetic expression	NOUN NOUN	3	-->	3
354	=:	lexical assignment	ADJ NOUN	3	-->	4
355	=:	phrase structure	NOUN	3	-->	45
356	=:	type assignment	NOUN NOUN	3	-->	5
357	=:	linguistic category	ADJ NOUN	3	-->	3
358	=:	proper treatment	NOUN_ADJ NOUN	3	-->	3
360	=:	higher order model	ADJ NOUN	3	-->	3
361	=:	non-logical constant	ADJ NOUN	3	-->	7
363	=:	arbitrary type	ADJ NOUN	3	-->	3
364	=:	type sound	NOUN NOUN_ADJ	3	-->	3
366	=:	identity function	NOUN NOUN	3	-->	4
368	=:	combinator scheme	NOUN NOUN	3	-->	4
369	=:	grammar rule	NOUN NOUN	3	-->	3
370	=:	grammatical theory	ADJ NOUN	3	-->	4
372	=:	beta eta long form	NOUN NOUN ADJ NOUN	3	-->	3
373	=:	induction hypothesis	NOUN NOUN	3	-->	3
....						
1395	=:	modal statement	NOUN_ADJ NOUN	1	-->	2
1406	=:	non standard logic	ADJ NOUN NOUN	1	-->	2
1456	=:	possible world model	NOUN NOUN	1	-->	2
1476	=:	first order modal logic	ADJ NOUN NOUN	1	-->	2
1520	=:	extensional semantics	ADJ NOUN	1	-->	2
1586	=:	context dependence	NOUN NOUN	1	-->	2
1653	=:	group reading	NOUN PROG	1	-->	2
1656	=:	selectional restriction	ADJ NOUN	1	-->	2
1697	=:	semantic type	ADJ NOUN	1	-->	2

[11] In the term list, strings like lambda, beta, eta, etc., are used to substitute for the Greek letters λ, β, η, etc., correspondingly, from the manuscript.

1717 =:	scope operator	NOUN NOUN	1 --> 2
1726 =:	moortgat's theory	NOUN NOUN	1 --> 2
1731 =:	polymorphic lexical entry	ADJ ADJ NOUN	1 --> 3
1772 =:	strict reading	ADJ PROG	1 --> 2
1806 =:	variable binding	NOUN_ADJ PROG	1 --> 5
1830 =:	embedded subject	PASTPART NOUN_ADJ	1 --> 2
1831 =:	matrix subject	NOUN NOUN_ADJ	1 --> 2
1860 =:	non indexical pronoun	ADJ ADJ NOUN	1 --> 2
1862 =:	indexical pronoun	ADJ NOUN	1 --> 3
1882 =:	logical operator	ADJ NOUN	1 --> 3

Appendix B: Non-terminological Phrases Filtered Out By "following"

15 =:	following example	PROG NOUN	23 --> 25
47 =:	following sentence	PROG NOUN	14 --> 14
53 =:	following analysis	PROG NOUN	12 --> 12
54 =:	following lexical entry	PROG ADJ NOUN	12 --> 12
56 =:	following scheme	PROG NOUN	12 --> 12
134 =:	following kind	PROG NOUN	6 --> 6
233 =:	following contrast	PROG NOUN	4 --> 4
238 =:	following clause	PROG NOUN	4 --> 4
359 =:	following definition	PROG NOUN	3 --> 3
362 =:	following assumption	PROG NOUN	3 --> 3
414 =:	following pair	PROG NOUN	3 --> 3
452 =:	following condition	PROG NOUN	3 --> 3
499 =:	following valid formula	PROG ADJ NOUN	2 --> 2
532 =:	following reading	PROG PROG	2 --> 2
556 =:	following dowty	PROG NOUN	2 --> 2
588 =:	following pattern	PROG NOUN	2 --> 2
618 =:	following postulate	PROG NOUN	2 --> 2
623 =:	following expression	PROG NOUN	2 --> 2
634 =:	following semantics	PROG NOUN	2 --> 2
640 =:	following form	PROG NOUN	2 --> 2
641 =:	following situation	PROG NOUN	2 --> 2
648 =:	following judgement	PROG NOUN	2 --> 2
732 =:	following logical equivalence	PROG ADJ NOUN	2 --> 2
783 =:	following collection	PROG NOUN	2 --> 2
826 =:	following formula	PROG NOUN	2 --> 2