A practical comparison of different filters used in automatic term extraction

Le An Ha

Research Group in Computational Linguistics
School of Humanities, Languages and Social Sciences
University of Wolverhampton
Stafford Street, Wolverhampton,
WV1 1SB, UK
L.A.Ha@wlv.ac.uk

Abstract

This paper discusses an experiment where different filters used in automatic term extraction (ATE) are practically compared. In the experiment, 8 filters, belong to three groups (lexical syntactic, statistical and semantic filters), are used to extract terms from two corpora from the domain of chemistry and of cancer research. The performance of each individual filter, and similarity among them are calculated. The experiment shows that: 1) simple filters maybe very efficient ones; 2) those filters are really different from each others; 3) the choice of which filters to be used is a domain, genre, and application-specific issue.

Introduction

Automatic term extraction plays an important part in natural language processing, especially recently, when, together with the explosion of internet, the amount of specialised texts which are electronically accessible is increasing exponentially, new concepts, and thus, terms are introduced with a speech that without automatic methods, we soon cannot cope with this information overload. As an illustration, within one year, nearly one hundred thousands new concepts have been accepted into UMLS knowledge sources, together with them are two hundred thousands new names. Different NLP applications, such as automatic summarization, automatic indexing, computer-aided terminology processing, ontology building, etc., rely on a good ATE component, to deal with this increasing domain-specific lexical data.

But the task of designing and implementing of an ATE component for a particular application is not an easy one. Although various automatic term extraction methods have been introduced in recent years, each is set to perform a specific task, use different kind of data, pre-processing tools, and is evaluated using specific schemes and measures. Thus there is nothing to guarantee that, if one re-implements a known system for her/his application, it will perform well.

One can also choose not to re-implement every components in a known system, but pick-up different components across different available ATE systems that (s)he thinks will perform well in her/his application, but a question remains: which one to pick up?

Generally say, an ATE system usually contains three main components, which are a lexical syntactic filter[1], a statistical filter, and more recently, a semantic filter. It should be noted that, "filter", in this paper, is a very general term. Different authors use different methods to "filter" good terms out of term candidates using different strategies, (i.e. a yes/no strategy and/or a weighting/scoring strategy).

It is also noted that, different ATE systems can use the same (or very similar) individual filters, only the combination are different. Also some filters can look very different, where in fact, they underline the same phenomenon, or produce similar results. This contributes to the difficulty of choosing filters that would be good for a specific application, i.e. if the use of a shallow parser do not significantly improve the performance of the lexical-syntactic filter, one will have to think carefully before buying/using one in ATE tasks.

There are a few state-of-the-art reviews in the field of ATE (Castellvi et. al. 2001, EL Hadi et. al. 2001, Kageura and Umino. 1996, etc.), which are very useful for researchers, but each seems to have its own problems. Most of them are descriptive reviews, i.e. they describe different methods rather than compare them. (Actually, comparative information sometimes did exist in those reviews, but only from a methodological viewpoint, not from a practical viewpoint, like which filters are similar to others, or which one performs well on which type of data, etc. The lack of real comparative data does little to ease the difficulty of implementing an ATE component.

Although El Hadi et. al. 2001 already tries to create a competition among various ATE system[2], we believe this is the first paper to compare different filters commonly used in ATE, to provide practical information, and to help researchers decide which filters they should adopt for their own applications.

1. Filters used in ATE

As mentioned in the previous section, an ATE (sub)system generally contains a lexical syntactic filter, a statistic filter and a semantic filter. In this section, we will discuss the assumptions behind those filters, and their natures in details. Future filters, such as discourse filters will also be discussed in brief.

[1] Some authors call this "linguistic filter", but this will become ambiguous when people begin to introduce "semantic filter" into ATE system. Theoretically, semantics is a brand of linguistics, thus "semantic filter" should be a part of "linguistic filter". In order to distinguish between traditional "linguistic filter" and the new "semantic filter", we will use the term "lexical-syntactic filter" for filters that use lexical and syntactic knowledge (part-of-speech tags, shallow parser information etc.).

[2] In that paper, for an unknown reason, the authors do not give any details about the results at all. It makes the paper not as perfect as it should be, and giving little advice to help a decision to choose among those methods.

1.1. Lexical syntactic filters

The underlining assumption of the use of lexical-syntactic filters in ATE is that terms, as lexical units, should have certain distinct lexical-syntactic features comparing to other units in texts. By discovering those lexical-syntactic features, and after that, using them, we can (partly) separate terms from other lexical units.

Theoretical research in the field of terminography often leads to descriptions of lexical-syntactic properties of terms, such as how they are constructed from prefixes suffixes, and root words, or how a term is formed from different lexical units (Ananiadou 1994). But while those researches are valuable for understanding the nature of terms, the coverage of those descriptions are limited, or in other words, when dealing with natural-occurring terms, it is inadequate. This may be due to the fact that, nowadays, people are very creative, and do not follow any standard in inventing new terms.

Practical lexical-syntactic features of terms, embraced by researchers in the field of ATE, on the other hand, seems to reveal little theoretical knowledge about terms, yet very efficient. For example, (Justeson and Katz 1996) describe terms using only one regular expression, [AN]*NP?[AN]*N, yet a very powerful one.

Lexical-syntactic filters can be divided into two groups. Group one uses only the word, lemma and part-of-speech information (for example LEXTER (Bourigault 1994), JUS (Justeson and Katz 1996)). The other group uses information provided by a shallow parser to identify terms (Arppe 1995, Hulth 2003). Both groups rely on an assumption that terms are basically noun phrases, and the main task is to identify noun phrases in texts. But will the use of a shallow parser help, in the field of ATE, is a question needs to be answered.

Strategies to identify domain-specific noun phrases in texts can also be classified into three approaches:

- Using term boundary markers (LEXTER)
- Using term part-of-speech sequences (JUS)
- Using shallow parsers (FDG (Tapanainen and Jarvinen. 1997) and LTCHunker (http://www.ltg-.ed.ac.uk/software/pos/)) to identify dependency information based on syntactic analysis.

In the first approach, the text will be split into maximal-length NPs, thanks to some lexical-syntactic patterns that clearly identify boundary of a noun phrase, such as verbs, pronouns etc., certain sequences of prepositions and determiner. Those boundary markers can be identified by empirical observation, taking advantage of negative knowledge about parts of terms, or by machine-learning methods, where a set of known terms can be used to "learn", or extract those markers.

In the second approach, instead of finding boundaries of terms, we will try to identify the probable part-of-speech (pos) sequences in which our terms will appear. Then every sequence of words that follows those pos sequences will be considered as terms. This approach, instead of negative knowledge as in the first one, uses positive knowledge.

The third approach makes use of shallow parsers, in different ways, to identify noun phrases in texts. Shallow parsers such as FDG, LTCHunker, CLARIT (Evans and Zhai 1996), etc. use different techniques of syntactic analysis to produce information used to identify noun phrases. For example, given that we have the sentence "*The Ea SOLID is the activation energy that has to be applied to any solid object to start a physical change.*" The LTCHunker returns chunks:

```
[[ The Ea SOLID ]] (( is )) [[ the activation energy ]] [[
that ]] (( has to be applied )) to [[ any solid object ]] (( to
start )) [[ a physical change ]].
```

and from this, noun phrases (*Ea SOLID, activation energy, solid object, physical change)* can be extracted.

With the same input, FDG shallow parser returns:

```
1     The    the    det:>3  @DN> DET
2     Ea     Ea     attr:>3 @A> N SG
3     SOLID  solid  subj:>4 @SUBJ N SG
4     is     be     main:>0 @+FMAINV V
5     the    the    det:>7  @DN> DET
6     activation    activation    attr:>7 @A> N SG
7     energy  energy  comp:>4 @PCOMPL-S N SG
8     that   that   subj:>9 @SUBJ PRON
9     has    have   mod:>7  @+FMAINV V
10    to     to     pm:>11  @INFMARK> INFMARK>
11    be     be     v-ch:>12      @-FAUXV V
12    applied apply  obj:>9  @-FMAINV EN
13    to     to     ha:>12  @ADVL PREP
14    any    any    det:>16 @DN> DET
15    solid  solid  attr:>16      @A> A
16    object  object  pcomp:>13     @<P N SG
17    to     to     pm:>18  @INFMARK> INFMARK>
18    start  start  mod:>16 @-FMAINV V
19    a      a      det:>21 @DN> DET SG
20    physical      physical      attr:>21      @A> A
21    change  change  obj:>18 @OBJ N SG
```

and more or less same noun phrases can be extracted.

It should be noted that, the three approaches can be combined in different ways, in order to give better results, for example, noun phrases provided by LTCH are put through pos sequence filters, or boundaries can be used to refine output from FDG. In fact, different ATE approaches usually combine those strategies, but in this paper, we test each strategy separately, in order to have a better view on the issue of which filters should be used for a particular application.

1.2. Statistical filters

The use of statistical filters in ATE is based on another assumption about terms in context, that the use of a term in a domain-specific corpus should be statistically different from those of other lexical units. Those statistical properties can vary from very simple ones (i.e. frequency) to very complex one (where the formula can be very long and complicated). But there is a suspicion that the complicated one is not always the best one, given that we often have to deal with small-scale corpus, where the terms do not appear frequently enough for the probability approximation used in those formulas to be reliable.

Statistical scores can be divided into two groups. Group one measures the use of the whole unit comparing to those of other units in the same corpus or in other corpora, such as term frequency, tf.idf, relative frequency etc. This reflects the assumption that a unit whose occurrence is biased in some way in a (document/domain) is likely to be a term. Group two measures the association strength among subunits (words) in a term. This is to find out whether the association between *physical* and *change* in "*physical change*" is strong enough to consider it a unit (term), or the appearance of "*physical change*" is just an accidental combination between *physical* and *change.* This association strength can be calculated using different measures, based on a standard contingency table. (Daille 1994).

Again, the above two approaches can be combined to measure both the usage and the association strength of the candidate, as in C-value (Frantzi and Ananiadou 1999).

1.3. Semantic filters

The use of semantic filters in ATE has been introduced recently, and more and more researchers are employing those filters for their ATE component. The idea is that semantic information should be used to identify terms, because, terms are specialized lexical units which have important meanings in their domains, thus should have certain semantic features which are different from other units. Example of approaches using semantic filters are (Maynard and Ananiadou 1999), where a semantic filter is encoded into context factor; (Paice and Black, 2003)'s three pronged approach; and Ha 2003a, 2004, where knowledge patterns are learned from resources and used for ATE.

1.4. Discourse filters

Discourse filters are considered to be a future direction for term extraction. For example, it is showed in (Ha 2003b) that performance of an ATE system can be improved by counting anaphoric expressions of term candidates. Other discourse theories, such as centering, and rhetoric structure may be useful for ATE. But the problem is that, current NLP techniques for the extraction of discourse information remain not reliable enough to be helpful.

2. Practical comparison of filters used for ATE

2.1. The setting of the experiment

In other to test different filters for ATE in a comparative way, we use following materials:

1) Corpus: we use to domain-specific corpora. One is articles collected from http://www.cancerhelp.co.uk website (CAN). Those articles contain general information about cancer and about different specific cancers, their diagnosing and treating methods, and other relevant information. The level of communication of those articles is from experts to users of intermediate knowledge about cancer. The size of this corpus is about 430000 words. The other (CHEM) is different articles of chemistry for beginners, contains about 350000 words. The source of those articles is also Internet.

2) Pre-processing tool: for filters that only use word, lemma and pos information, FDG shallow parser will be used to provide such information. Furthermore, LTCHunker is also used to test the use of shallow parser.

3) Training set of terms: there are no official training set of terms, but in case a filter needs, a small set of known terms from a domain-specific glossary will be provided.

4) Testing data: testing data will be the terms provided by using domain-specific glossaries, from the website of cancer research itself, and a chemistry glossary.

It is known that full evaluation for ATE is very time-consuming. A full analysis of every term candidate extracted by the system will be required, to confirm or deny the term status of the candidate. Furthermore, different analysis strategies will lead to different results. In this experiment, no manual analysis will be performed, but a list of known terms will be used. Thus the number of "correct" terms is only calculated against the known list of terms, and cannot be considered as an absolute figure. It should be used only as a reference point for comparing different methods.

2.2. Filters to be tested.

We choose filters to represent different type of filters. For lexical syntactic filters: Justeson and Katz (JUS) regular expression; LEXTER boundary markers, LTCHunker and FDG. For statistical filters, we choose term frequency, mutual information (MI) and C-value[3] to represent three types of statistical measures (see section 1.2). For semantic filter, only Ha approach is tested, due to the fact that it is available to our research.

2.3. Implementation notes

Both JUS regular expression and LEXTER boundary marker filters are very easy to implement, and we can use any part-of-speech tagger as pre-processing for them. LTCHunker is freely available for research purposes, and FDG shallow parser is a commercial tool. All the statistical scores are easy to calculate, and Ha's semantic filter requires syntactic information from FDG parser. For lexical syntactic filters, we extract every lexical unit that satisfies the filter. For statistical filters, we extract the first 1500 highest score units. Similarity between filters are calculated using cosine distances (number of identical units/(square root of (total units from filter 1 multiplied by total units from filter 2).

Given that only one semantic filter has been used, we calculate the improvement when the semantic filter is applied.

2.4. Results

Lexical-syntactic filters

Table 1 compares the total number of lexical units identified by different lexical-syntactic filters, and the number of correct ones (see section 2.1). Results from the table suggest that, in the field of ATE, the use of shallow parser does not guarantee higher accuracy. And JUS regular expression seems perform very well across the domains. Table 2 shows the similarity among those filters. It shows that those lexical-syntactic filters are really different from each other, thus one has to choose one of those filters very carefully, in order to extract the right lexical units they want. The fact that JUS and LTCH is the most similar pair implied that the grammars used by them maybe similar.

[3] We choose simple ones, because their natures are better understood, and they are easy to implement. It is also shown in different works that a simple measure is not necessary a poor one.

	LEXTER		JUS		LTCH		FDG	
	#t	#c	#t	#c	#t	#c	#t	#c
CHEM	29600	453	24834	509	18364	506	37572	507
CAN	32676	913	19604	991	19412	940	33901	935

Table 1: Number of lexical units identified by different lexical syntactic filters (#t) and the number of those units which are identical to the one from a glossary (#c).

	LEXTER-JUS	LEXTER-LTCH	LEXTER-FDG	JUS-LTCH	JUS-FDG	LTCH-FDG
CHEM	0.49	0.35	0.48	0.54	0.48	0.38
CAN	0.46	0.37	0.46	0.53	0.47	0.39

Table 2: Similarity between those lexical syntactic filters

Statistical filters

Table 3 shows how representatives of different types of statistical filters perform. As it shows, the performances of different statistical filters vary from domain to domain. It is not necessary that a certain statistical filter should perform well across every domain, again, suggested that the decision to choose a statistical filter for a specific application is not an easy one, and one will have to take into account different factors. The similarity between different pairs (table 4) also suggests that, they are not similar to each other.

	Fre	MI	C-Values
CHEM	0.48	0.49	0.50
CAN	0.40	0.35	0.31

Table 3: Percentage of correct terms among the first (roughly 1500) term candidates

	Fre-MI	Fre-C_value	MI-C_Value
CHEM	0.40	0.48	0.50
CAN	0.38	0.48	0.54

Table 4: Similarity between those statistical filters

Semantic filter

The use of a semantic filter (see section 2.1) is shown to improve the accuracy of ATE by around five to nine percent, which suggests that the use of semantic filters should be taken into account for future ATE systems.

3. Conclusion and future works

During the experiment, there are several lessons that have been learned. Firstly, in the field of ATE, there is no filter are shown to be universally good, each domain may require a different filter, and choosing a filter is an empirical issue. Secondly, the use of complicated techniques (i.e. shallow parsing, complex statistical measures) does not guarantee the improvement over the performance. Part-of-speech pattern technique is shown to be the best one in this experiment, and term frequency is not always the weakest one among statistical filters.

To build an ATE system, one should use three filters, namely lexical syntactic, statistical and semantic ones. And within each type of filters, choosing the right one will depend on various factors, including domain, genre, application, etc.

Future works will include an investigation into whether or not a combination between different filters, using optimization and machine-learning techniques, will improve ATE performance, and an extensive investigation of different semantic filters.

References

Ananiadou, S. 1994. “A methodology for Automatic Term Recognition”. *Proceedings of COLING94*, 1034-1038.

Arppe, A 1995. “Term extraction from unrestricted text.” *Lingsoft website: http://www.lingsoft.com.*

Bourigault, D. 1994. “LEXTER, un Logiciel d’Extracton de TERminologie des connaissances a partir de textes”. PhD Thesis. Paris: Ecole des Hautes Etudes en Sciences Sociales.

Castellvi, M. T. C., R. E. Bagot and J. V. Palatresi. “Automatic term detection, a review of current systems.” *Recent Advances in Computational Terminology* ed. by Bourigault, D., Jacquemin, C. and L’Homme, M. John Benjamins, 53-87.

Daille, B. 1994. “Approche mixte pour l’extraction de teminologie: statistique lexicale et filtres linguistiques.” PhD thesis. Paris: Universite Paris VII.

Frantzi, K.T and S. Ananiadou. 1999. “The CValue /NC-Value domain independent method for multi-word term extraction”. *Journal of Natural Language Processing*, 6(3):145-179.

Kageura, K. and B. Umino. 1996. “Methods of Automatic Term Recognition, a review”. *Terminology*. 3(2), 259-289.

Ha, L. A. 2003a. “Extracting important domain-specific concepts and relations from a glossary”. In *Proceedings of the 6th CLUK Colloquium.*

Ha, L. A. 2003b. “Do we correctly count the term frequency? The influence of the anaphoric expression of terms in automatic term extraction.” In *the Proceeding of RANLP 2003.*

Ha, L. A. 2004 . “Co-training applied in automatic term extraction”. In *Proceedings of the 7th CLUK Colloquium.*

El Hadi, W. M., I. Timimi, A. Beguin and M. de Britto. 2001. “The Arc A3 Project: Terminology acquisition tools: Evaluation Method and Task”. *Proceedings of the ACL 2001 Workshop on Evaluation Methodologies fro Language and Dialogue Systems.*

Hulth, A. 2003. “Improved Automatic Keyword Extraction Given More Linguistic Knowledge”. In *the Proceeding of RANLP 2003.*

Justeson, J. S. and S. L. Katz, 1996. “Technical terminology: some linguistic properties and an algorithm for identification in text”. *Natural Language Engineering*, 3(2), 259-289.

Maynard, D. and S. Ananiadou, 1999. “Identifying Contextual Information for Multi-Word Term Extraction”. *Proceedings of the Fifth International Congress on Terminology and Knowledge Engineering (TKE'99)*, 212-221. Vienna, Austria.

Paice, C.D. and W.J. Black. 2003. “A Three-pronged approach to the extraction of key terms and Semantic Roles”. In *the Proceeding of RANLP 2003.* 357-363.

Tapanainen, P. and T. Jarvinen. 1997. “A non-projective dependency parser”. *Proceedings of the 5th Conference of Applies Natural Language Processing* .64–71.