Co-training applied in automatic term extraction: an experiment.

Le An HA
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 the use of a setting similar to co-training in automatic terminology processing. Two aspects of terms (internal aspect, i.e linguistic, and statistical properties; and external aspect, i.e. contexts) will be used interchangeably in a bootstrapping manner, in order to extract more and more terms and context patterns. The results show that, using only a small set of seed terms, the method can extract terms, with higher success rates than those of other methods. Further more, this method can also discover interesting context patterns, which can be used in other terminology processing applications.

1 Introduction

Automatic terminology processing has proved to be a necessity, when, together with a large number of scientific achievements, lightning-fast scientific communication using the Internet, new concepts, and terms, are introduced with an exponential speed. For example, within one year, 98,315 (10%!) more concepts, and 226,729 more names are added into UMLS Knowledge Sources(http://www.nlm.nih.gov/research/umls/) Only automatic processes can help us handle this increasing amount of textual data from scientific communication.

In automatic terminology processing, it is not only important to extract terms[1], but also patterns that may reflect the meanings, or"position", of those terms in a terminology.

In the search for methods of automatic terminology processing, co-training-like settings appear to be promising ones. Exploring an assumption that there are two different, independent aspects of the linguistic phenomenon "term", using only a very small set of seed terms, the co-training setting can be used to extract instances of each aspect using the other one interchangeably and incrementally. In this paper, we will set up such an experiment, using "internal" term properties, such as part-of-speech sequence and statistical scores, and "external" term properties, i.e. context patterns as two independent aspects of "terms", each of which will be used to extract more instants of the others, and the process will be then repeated for n iterations. Through this experiment, we try different "internal" and "external" properties, to identify which ones are more suitable for our purpose (which is to extract domain-specific terminology). The experiment shows that, among the 11 pattern heuristics introduced by (Riloff 1993), only 4 of them are suitable for our work. As a result, we introduce one more pattern heuristic, exploiting the output of FDG super tagger, "<noun_phrase> that/which verb". This heuristic is proved to be very useful and productive. Further more, in addition to the <subject> active_verb pattern, which is sometimes too general, we use <subject> active_verb supplement instead, and the same with <subject> passive verb.

The experiment also shows that, when combine with other statistical scores, the method can improve the performance of general-purpose

[1] In this paper, we consider terms as linguistic labels of domain-specific concepts.

automatic terminology processing methods. Not only terms are extracted, knowledge patterns, which can be used for further application of automatic terminology processing, are also among the context patterns identified by the proposed method. It also shows that the use of such general pattern heuristics can be applied for different domains. We try our method on two domains (chemistry and cancer research) to assets the domain-independent nature of the proposed method.

2 Related work.

There are two main branches of research which relate to this paper, one is co-training like settings, and the other is automatic terminology processing. The two will be discussed below.

2.1 Co-training-like settings

From the day the term co-training setting[2] was introduced (Blum and Mitchell 1997), more and more people are using them to solve different natural language processing problems. The methodology is widely used mainly because of its nature, which is to only use a small amount of annotated texts, then combine with a larger amount of un-annotated data and two fairly independent sets of features to "train" two separated classifiers, in a bootstrapping manner. At each round, more instances of each set of features are extracted using the previously trained classifier from the other current set of features. The process then is repeated for n iterations, and the number of instances extracted at each round is increased accordingly.

(Riloff and Jones 1999) classify noun phrases as negative or positive examples of locations, using the noun-phrase itself and the linguistic contexts as two sets of features. (de Sa and Ballard, 1998) use audio signal and video signal watching the speaker's lips to cluster corresponding to the spoken phonemes in the data. (Nigam et. al. 2000) use a co-training setting to combine an Expectation Maximisation and a Bayes classifier in the task of text classification. Those are a few examples of the use of co-training settings in different tasks.

[2] In this paper, we share the classification of Mitchell when considering mutual bootstrapping methods proposed by Riloff as co-training setting.

2.2 Automatic terminology processing

Recently, it is agreed that automatic terminology processing should not only be limited to extracting terms, but also other important information that relates to terms. In other words, the ultimate objective of automatic terminology processing is to build a terminology, which will include not only terms but also their descriptions, their friend terms, and their relations. Toward this direction, we can have work of (Meyer et. al. 2001), which showed that we could build a terminological knowledge based using different types of "knowledge patterns". Sharing the same idea of "knowledge patterns", but using different notions, (Paice and Black 2003) introduce a "three-pronged" approach, which combines linguistic, statistical and semantic features to extract terminology. In that work, semantic features are roughly equivalent to "knowledge patterns" as in the work of Meyer et. al. The bottleneck of this direction is that, "knowledge patterns" are still to be identified manually, thus making it a time-consuming, and domain-specific task. To solve this problem, (Ha 2003) suggested that we can look at textual data that are knowledge pattern-rich (i.e. glossaries), in order to quickly extract those necessary patterns.

Another direction is works by Ananiadou, Frantzi, Maynard and others on term clustering, in which the "position" of a term in a terminology is identified by its "friends". In those works, extracted terms are clustered into groups, which can reveal the relations between different terms in the domain. If combined with the "knowledge pattern" approaches discussed above, we can expect to have a better representation of the "position" of an individual term in the terminology. In such a terminology different useful information about terms, such as information about their properties, their friends and their relations to those friends are presented.

3 The algorithm and its settings

3.1 The algorithm:

Similar to the one suggested by (Riloff 1999), we propose the following algorithm to extract terms and context patterns:

1) i=0; curNumContextRule=m; curNumTerm=n;
2) Choose an initial set of terms as seeds. T_0
3) Extract every context patterns around those Ti: Ci=C(Ti)
4) Sorting those Ci according to a sorting function. Assign Ci = first curNumContextRule ones of this sorted list.
5) Extract every term candidates using Ci: Ti+1=T(Ci).
6) Sorting those Ti+1 based on a sorting function. Assign Ti+1=first curNumTerm.
7) I=I+I;curNumContextRule+=incrRule; curNumTerm+=incrTerm;if(I=maxIteration) goto 8, else goto 3.
8) Print out context rules and terms.

The above algorithm looks simple but contains several problematic issues, which we will discuss below.

3.1.1 Context pattern heuristics

The first problem is the question of which context patterns we should use in this experiment, there is already one set of heuristics, proposed by (Riloff 1993), which contains eleven linguistic patterns, such as <subject> active_verb; <subject> passive_verb; noun_prep <noun_phrase>; verb_prep <noun_phrase>; etc (see the paper for the full list of those heuristics. A problem with those heuristics is that some of them are too specific, and some of the others are too general. For example, <subject> be(verb) is a very general, uninformative pattern.

Further more, there are still some overlooked patterns, such as the one with relative clauses. Observations show that relative clauses are widely used to describe concepts, and should therefore be considered as important patterns.

Another problem is that the set proposed by Riloff is not designed for terminology processing, but other lexical items, thus some patterns will not be useful.

Yet another issue that should be addressed is that of how, and where to get the required syntactic information, in some case when there is a distance between subject and verb? Luckily, the shallow parser we use provides such information, and this experiment (FDG shallow parser (Tapanainen and Jarvinen 1997)) can also be considered as an extrinsic evaluation of this parser.

3.1.2 Sorting function

Sorting functions for terms and context patterns can be considered as classifiers as in other co-training settings. Sorting functions play an important part in the proposed algorithm, because it will decide how well the process can discover more and more reliable instances of term candidates and context patterns. A poor sorting function will lead to very poor results, because of the bootstrapping nature of the algorithm. Because of the different natures of context patterns and terms themselves, there should be one distinct function for each of them. A function for context patterns will have to make sure that we can extract informative ones, whereas for terms, it (the function) should be closely related to other *termhood* measures.

3.1.3 Initial states and parameters

Initial states and parameters seem to be less important than other issues, but still, questions such as: how the process will react to different initial sets of terms and parameters, or how carefully one has to be to choose the initial seeds, should be addressed. Because of the nature of the algorithm, as showed in (Riloff 1999), that tends to extract items which are semantically close to the seeds terms, ideally such an initial set of terms should cover different semantic classes in the domain (such as chemical compounds, processes, methods, properties etc. as in the domain of chemistry).
The same question can be raised with other parameters, (i.e. incrRule; incrTerm; maxIteration). Preferably, the number of rules should increase more slowly than the number of terms, and the number of iterations has to be not too large or too small.

All the above three issues will be practically addressed in the next section, where details of the experiment are described.

4 The experiment, results and discussions

4.1 Domains, corpora, evaluation data

In order to perform the experiment, we have collected two sets of data, one is in the domain

of chemistry, where we collected different introduction-to-intermediate-level texts from the internet (the whole corpus contains about 350000 words); the other is in the domain of cancer research, where the texts come from the website: http://www.cancerhelp.co.uk (450000 words). The level of communication of those texts is from expert to reader of intermediate level of knowledge about cancer. The reason to choose these texts is that they contain more "textual data", and less "technical data" (i.e. figures, formulas, equations, etc.) which can not be processed by NLP techniques with high accuracy.

Within each domain, we collected a glossary, which can serve as evaluation data (as we will discuss below, this evaluation data can not be considered as gold standard, but only a reference point for the performance of the proposed method).

4.2 Evaluation scheme.

Evaluation is the bottleneck of automatic terminology processing. Expert opinions are expensive and time consuming. Using available resources is not a perfect solution. Available resources are often incomplete, or even faulty, for example, a chemistry glossary includes "carbon dioxide", but not "chlorine dioxide", whereas both should either be or not be considered as terms. Even carefully built terminologies, such as UMLS (http://www.nlm.nih.gov/research/umls/), have certain inconsistencies and errors. This dilemma leads us to a strategy of two-stage evaluation. In stage one, where we try different parameters optimising the performance, to ensure the speed of the process, we use the glossary and UMLS database as gold standards, to observe the effects of different factors relatively. After having the optimised set of parameters, human opinions and corpus evidence will be used in the final evaluation, where a term candidate can be considered as a correct term if there is corpus evidence that it appears in contexts that reflect the importance of the candidate in the terminology of the domain. (a chemical compound is used in some process, or a name of a type of cancer).

With regard to the patterns, we calculate the ranks of extracted patterns from this experiment using the score suggested by (Ha 2003), to observe the relation between the two unrelated methods.

4.3 Detailed setting of the algorithms

In this subsection, we will discuss the detailed setting of algorithm, and how we decide which ones to be used in the final version of the experiment.

4.3.1 Term-related issues

Our experiments show that 99% of the terms appear in both glossaries have part-of-speech sequences satisfying the regular expression ((A|N)+|(A|N)*(N P) (A|N)*) N, which is suggested by (Justeson and Katz 1995). Thus this pattern will be used to filter term candidates. Syntactical dependencies and functions given by FDG shallow parser will be used when syntactic information is needed. For example, given that we have the pattern <subject> contain, and notice the verb "contain" in a sentence, we will 1) find the noun which is marked as subject of "contain"; and 2) find the longest sequence of (not longer than 5 words) that contains this noun and satisfy Justeson's pattern. This item, then, will be considered as a term candidate.

For the purpose of sorting term candidates (see section (3.1.2), it is noted that other "termhood" functions may be useful. But due to the time restriction, we only try two simple ones, and in both cases, the performances are improved comparing to when we only use those termhood measurements alone. We hope that this behaviour will be the same with other termhood measurements. The two scores we use are 1) frequency and 2) C-Value (Frantzi and Annaniadou 1998). We only calculate those scores on and relative to the extracted term candidates. Table 3 compares the performance of the four settings.

4.3.2 Pattern-related issues

The investigation into which pattern heuristics should be used unearths some interesting observations. Firstly, the term/non-term discrimination power of certain general patterns, such as <*subject*> *be*; <*subject*> *have* etc. is very low, thus if we still want to use those pattern, we will have to extend them into more specific patterns, like <*subject*> *be*

object/complement (is stable; is a gas; etc). A systematic way to identify which patterns are too general is to run the program, and then identify patterns with significantly high frequencies, and try to construct more specific pattern heuristics from them. The fact that a pattern has a high frequency shows that may be we can benefit more from a more specific pattern rather than this general one.

Experiments also show that patterns like active_verb prep <noun_phrase>; passive_verb prep <noun_phrase>; passive_verb <object> actually reduce the performance rather than improve it. Those patterns are also eliminated from the final version of the program.

Pattern heuristics	Examples
1 : Noun Prep <Noun_Phrase>	property of…; treatment for…
2 : <Subject>active_verb	…. contain; …. include
3 : <subject> passive verb	…are used;…. is formed
4 : active_verb <object>	contain …; call …
5 : <noun_phrase> rel pron verb	… that is; … which contain
6 : <subject> active_verb supplement	… consist of; … is important
7 : <subject> pass_verb supplement	… is used in; … is defined as

Table 1 : pattern heuristics that are proved to be reliable and used in the final experiment.

With regard to relative clauses, we introduce one pattern, <noun_phrase> [that|which] verb. This additional pattern is proved to be useful for our task. The information from FDG parser allows us to identify the syntactic dependency between relative pronouns, verb and noun_phrase with an acceptable accuracy. A summary of pattern heuristics will be used in the final version of the program can be found in table 1; and table 2 show the performance of the system when using different combinations of pattern heuristics. (in table 2, the left-most column reflects the number of iterations and the total number of term candidates identified; for type of patterns in the first row see table 1).

	1	1+2	1+3	2+3	1+2 +3	1+2+ 3+4	1+2+3 +4+5
5:107	44	43	43	26	46	49	50
13:257	80	82	80	59	80	91	95
25:507	122	123	123	73	122	144	151
50:1007	176	177	176	96	176	208	227

Table 2: the effects of different pattern heuristics on the number of correct terms extracted.

A sorting function for patterns is a difficult issue. In some way, this sorting function acts as a "classifier" which identifies good patterns among pattern candidates. Unlike terms, where we have different "termhood" scores, we do not have any reliable "patternhood" measurements. Can we simply use frequency as a patternhood measure? Through the experiment, the answer is no, and the reason is that, a high frequency pattern is not always the informative and productive pattern. A sorting function for patterns should take into account that a pattern needs not to be abundant, but to be productive, and informative. Thus we should use an information-related function such as entropy, or the one suggested by (Riloff and Jones 1999). Our hypothesis is that a good pattern should be an informative one, which leads to the use of entropy as a sorting function. Thus the score for a pattern j is calculated as its entropy

$$H_j = \sum_{T_j} p_i \log(p_i)$$

where p_i=(the count of the term candidate i predicted by pattern j)/(total count of term candidates predicted by pattern j). T_j is the set of term candidates predicted by the pattern j. This formula is more general than the one used by Riloff, with only the number of term candidates predicted rather than the distribution of them being used. This score favours patterns that appear around a large number terms indiscriminately rather than the ones only appear (heavily) around a small number of terms.

4.3.3 Initial seeds and other issues

From experiments, we get an impression that we do not have to choose the initial seeds very carefully. We only have to make sure they cover different semantic classes in the domain. It is also showed that, when the number of seed terms is increased, the performance does not improve significantly, which suggests that the algorithm, in this situation, is, in fact, converse. Because of the size of our corpus, it is not advisable to use too many iterations. We also set the **incrRule** at 5 and **incrTerm** at 20. Seed terms for chemistry are: *carbon dioxide, resonance stabilization, acidic solution, atom, bleach, electron, crystallization, hydroxide, covalend bond, ionic compound, qualitative analysis, absorption spectrum, triglyceride,*

intermolecular force, compound, nitrogen; and for cancer research are: *abdominal radiotherapy, aminoglutethamide, antiangiogenic drug, cyclophosphamide, liver cancer, lymphokine, placebo effect, second cancer, chemotherapy, endoscopy, follicular mixed cell, relaxation exercise, ultrasound scan, affected lymph node, bowel motion.*

4.4 Final Results

After fine-tuning the parameters of the algorithm accordingly, we run the program on the two corpora described in section 4.1, and the results are showed in tables 3 and 4.

In Table 3, we compare the accuracy of extracted term candidates (using the evaluation scheme discussed in section 4.2) of methods: 1) only using frequency; 2) using Cvalue; 3) co-training which uses frequency as the sorting function for term candidates, and 4) co-training which uses Cvalue, in the two domains. As we can see, the use of the co-training setting improves the performance in both cases, both domains. The improvement is about 6 to 8% of the accuracy rate.

		Fre	**CValue**	**Fre+ cotrain**	**Cvalue+ cotrain**
chem	#c	782	850	957	980
	#total	1341	1395	1439	1420
	%	58	60	66	69
cancer	#c	1239	987	1331	1124
	#total	2484	2375	2323	2401
	%	50	41	57	46

Table 3: final results (#c: the number of correct terms extracted).

Table 4 shows patterns extracted and best term candidates identified by them, the first number in brackets show the pattern type (see table 1); and the last number in brackets is the rank of the pattern using the method suggested by (Ha 2003). The patterns identified by the method, in both domains, are somehow similar to the knowledge patterns extracted by Ha's method. This suggests that, patterns identified by the method proposed in the paper are not just patterns, but may also contain domain knowledge, and can be used in future automatic terminology processing applications.

5 Conclusion, and future direction

The experiments show that, with some adjustments, the co-training (or mutual bootstrapping) settings can be applied in the field of automatic terminology processing. In fact, it improves the success rate of automatic term extraction, and can be used to tackle problems of identifying both terms and knowledge patterns for further applications. But there are a lot of issues still to be addressed, both in the co-training setting itself and in this particular application of the setting. For example, which will be the optimised classifiers for this application. In the experiment, we use certain sorting functions as classifiers, but will other classifiers work better? Another issue is whether the set of pattern heuristics used is optimised, or some heuristics will have to be more specific/generalised. Yet another issue that has not been addressed in this paper is a mechanism to have more "control" of the extracted patterns and term candidates at each round, something like meta-bootstrapping, which, in the end, is about designing better sorting functions for both terms and patterns. Those are some of the possible future directions the research will take.

Acknowledgement

Thanks to the members of my research group, who provide valuable comments at the early state of this paper, and also to the anonymous reviewer, who suggested various interesting ideas.

References

Blum, A. and Mitchell, T. 1998. "Combining labeled and unlabeled data with co-training". In *Proceedings of the Workshop on Computational Learning Theory*. Morgan Kaufmann Publishers.

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.

Ha, L. A. (2003). "Extracting important domain-specific concepts and relations from a glossary". In *Proceedings of the 6th CLUK Colloquium*, 6 - 7 January, Edinburgh, UK, pp. 49 – 56

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.

Meyer, I. 2001. "Extracting knowledge-rich contexts for terminography: A conceptual and methodological framework". *Recent Advances in Computational Terminology* ed. by Bourigault, D., Jacquemin, C. and L'Homme, M. John Benjamins, 279-302.

Nigam, K. McCallum, A. Thrun, S., and Mitchell. T. 2000. "Text Classification from Labelled and Unlabelled Documents using EM.". *Machine Learning.* 39(2/3). 103-134.

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

Riloff, E. and Jones, R. (1999). "Learning dictionaries for information extraction using multi-level bootstrapping". In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*. The AAAI Press/MIT Press.

Riloff, E. 1993. "Automatically constructing a dictionary for information extraction tasks". In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, 811--816.

de Sa, V. and Ballard, D. 1998. "Category learning through multimodality sensing". *Neural Computation*, 10(5). 1097-1118.

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

Chemistry	**Cancer research**
which be (5) (best terms: orbital, density distribution, electron configuration ...) (6)	that be (5): treatment, cancer, lymph node, tumour, genetic code (2)
contain (4): particle, neutron, oxygen, element, anion, visible light. (1)	treatment for (1): breast cancer, cancer, abnormal smear, prostate cancer, Hodgkin disease, bowel cancer (32)
be in (6): chlorine, outer electron, nitrogen, ion, sodium, particle, oxygen. (148)	information about (1):specific side effect, different type*, survival rate, ct scan, mri scan (-)
that be (5): solution, element, reaction, substance, molecule (6)	symptom of (1): brain tumour, lung cancer, secondary breast cancer, advanced cancer (40)
represent (4): mole, ion, layer of plane, concentration, mass, molecule in equation, repulsion (50)	be likely (6): cancer, doctor, woman, treatment, high grade cancer, (159)
concentration of (1): hydrogen ion, charge density, weak acid, magnesium nitrate, saturated solution (17)	use (4): wave, system, painkiller, magnetism, x-ray (21)
amount of (1): energy, solute, reactant, charge density, electron density, electron charge, phosphate ion (2)	be a type (6): ct scan, malignant melanoma, humeral replacement surgery, lymphoma (31)
mole of (1): solute, gas, hcl, water, nitrogen, gas particle, acetylene, solute per liter, sodium nitrate (3)	treatment with (1): bone marrow, 5fu, trastuzumab, doxorubicin, (13)
change in (1): concentration, state, oxidation state, dipole moment, (6)	risk of (1): breast cancer, infection, severe infection, bacterial infection. (6)
property of (1): polymer, solution, matter, chlorine dioxide, ionic compound. (62)	effect of (4): chemotherapy drug, individual drug, combination of drug, cancer treatment, 5fu (4)

Table 4: patterns extracted and best term candidates identified by them