Extracting Technology and Effect Entities in Patents and Research Papers

Jingjing Wang, Han Tong Loh, Wen Feng Lu
Department of Mechanical Engineering, National University of Singapore, Singapore
{wang_jingjing, mpelht, mpelwf}@nus.edu.sg

ABSTRACT

This paper describes our approach to tackling the task of Technical Trend Map Creation as posed in NTCIR-8. The basic method is Conditional Random Fields, which is considered as the most advanced method in Named Entity Recognition. In order to improve the performance, we further resort a tag modification approach and pattern-based method. Our system performed competitively, achieving the top F-measure among participants in the formal run.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *text analysis*

I.5.4 [**Pattern Recognition**]: Applications – *text processing*

General Terms

Algorithms, Performance, Design, Experimentation

Keywords

NTCIR, patent mining, CRFs, patterns

1. INTRODUCTION

The patent mining task of NTCIR-8 opened a very interesting subtask. The purpose of the Technical Trend Map Creation task is to extract expressions of element technologies and their effects from research papers and patents. The product is obviously very useful for many applications, especially for technical trend map creation.

There is no strict definition of all specified terms consisting of TECHNOLOGY, EFFECT, VALUE and ATTRIBUTE. However, TECHNOLOGY is described as algorithms, tools, materials, or data used in each study or invention; EFFECT includes one or more pairs of ATTRIBUTE and VALUE. Examples of effects of a technology that are expressed by a pair of an attribute and a value are shown as follows:

$\{[\text{reduce}]_{\text{VALUE}}\ [\text{the manpower}]_{\text{ATTRIBUTE}}\}_{\text{EFFECT}}$

$\{[33\%]_{\text{VALUE}}\ [\text{redundancy-rate}]_{\text{ATTRIBUTE}}\}_{\text{EFFECT}}$

According to our observation, a "technology" or "attribute" is usually a noun or noun phrase, and a "value" can be a verb, gerund, adjective or a number.

The whole evaluation process has two rounds i.e. a dry run and a formal run. For each round, the organizer of NTCIR-8 offered tagged topics for training and untagged topics for test. The raw text of each topic is the title and the abstract of a patent or a paper. We call a topic as patent (or paper) topic if its raw text is from a patent (or paper). The evaluation, which is based on recall and precision, was executed by the organizers.

This is our first time to attend NTCIR task and we participate both dry run and formal run. After the dry run, the tagged test data for dry run was released. Therefore, our system was optimized using the dry run data.

2. RELATED WORK

This Technical Trend Map Creation task seems a continued effort to generate patent matrix map, since NTCIR-4 organized a similar task [1]. However, the results in NTCIR-4 were not satisfied [2, 3]. A similar but more detailed work [4] was done in 2007. Nevertheless, this Technical Trend Map Creation task is different from above works, because it does not be limited to patent, and moreover, it seems easier i.e. it does not require a matrix map generation step.

On the other hand, this Technical Trend Map Creation task is more like a typical Information Extraction (IE) task [5] or Named Entity Recognition (NER) task, although the desired information snippets are extraordinary. Early works in IE or NER were pattern-based with manually coded patterns or automatically learned patterns [6, 7]. Then came the age of statistical learning, in which Conditional Random Fields (CRFs) [8] was considered as the state-of-the-art method for assigning labels to token sequences [5, 9]. Such statistical method does not rely on patterns, which are too brittle in a noisy source. However, it does not mean that statistical method is better than pattern-based method. There also exist hybrid systems [10] that attempt to obtain the benefits of both methods.

In this Technical Trend Map Creation task, two critical issues should be (1) finding effective and efficient features to highlight the desired information snippets from plain text, and (2) building an advanced model to sufficiently utilize all these features. Moreover, we hoped to obtain an acceptable performance, so that the discovered information snippets would be used for more advanced task.

Therefore, we started from a CRFs based method i.e. an advanced model with many features. In order to improve the performance, we did a slight modification on the original CRFs model. Since the output results were still not satisfied, we further added some patterns and invoked a pattern-based method.

3. SYSTEM OVERVIEW

The architecture of the system, which includes three output modules, is shown in Figure 1. This section introduces the processes from the input topics at the top right of the Figure 1 to the first output, namely output 1. Tag modifier and pattern-based extractor will be introduced in next two sections.

Patent and paper topics were firstly separated. Obviously, there exist intrinsic differences in writing custom between these two kinds of style; and it is too difficult to make sure such differences, which are considered as noises. Mover, there also exist another noise, which is the slight difference in representing HTML characters; for example, α (character code: 03B1 from Unicode-hex) is “&alpha;” in paper, but is written as “.alpha.” in patent. A check list that links all HTML characters in patent topics with that in paper topics can eliminate the effect of the second kind of noise, but to build such a check list may be not a good idea. In order to simplify the problem, we chose to distinguish patent and paper topics instead.

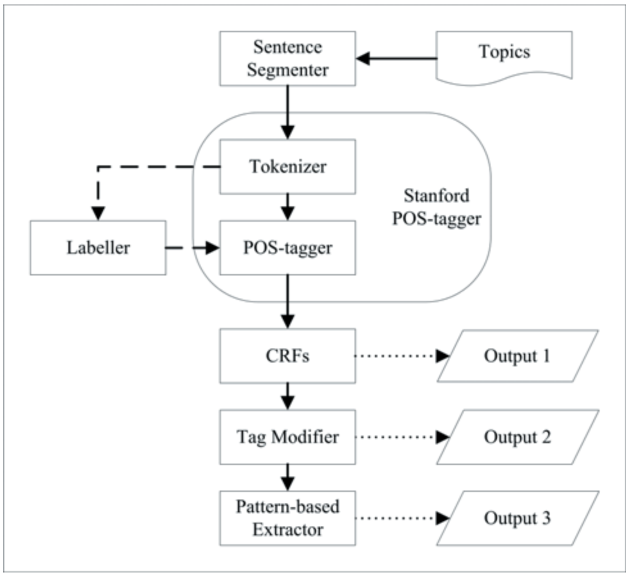

Figure 1. System overview

3.1 Sentence Segmenter

Our sentence segmentation techniques are robust to HTML characters in patent. As above example, α is represented as “.alpha.” in patent; the two dots in “.alpha.” obviously cause a problem in sentence segmentation. Furthermore, it can cause a problem in tokenization. Therefore, the sentence segmenter deletes the dot, when it belongs to a HTML character.

Moreover, the sentence segmentation techniques adapted are intelligent to many language situations such as suspension points, abbreviation, paper number, decimal value. For examples, dots in “i.e.”, “vs.” and “7.654” are not considered as periods.

3.2 Tokenizer and POS-tagger

We use Stanford pos-tagger [11, 12] for both tokenization and POS-tagging. Since the Stanford pos-tagger was adopted for pos-tagging and the default pertained model of the Stanford pos-tagger was used, it is better to tokenize the raw text in the same manner, namely Penn-Treebank-based tokenization, as that used by the pertained model.

3.3 Labeller

The labeling scheme is BIO (begin, inside, outside), which is commonly used. We used three kinds of positive tags i.e. “technology”, “value”, and “attribute”, and one kind of negative tag i.e. “other”; each positive tag can be either “begin” or “inside”. So there are totally seven types of tags.

Since the tagging scheme in the training data is two levels, we had tried a hierarchical labeling scheme, which also has two levels. In the first level, tags include “technology”, “effect” and “other”; in the second level, only observation sequences with “effect” tags are considered, and tags involve “value”, “attribute” and “other effect’. Operation in this manner makes the training and test more complex, because it leads to hierarchical training and test. During the training, a model is trained using the tagging of the first level, and another model is afterwards added using the tagging of second level; for test, “technology” and “effect” is firstly extracted using the first model, then second model is used for extracting “value” and “attribute” in the “effect” just obtained in previous step. If there is nothing obtained from the first step, then the second model is entirely useless. In the experiments, this hierarchical labeling scheme demonstrated a very bad performance. Therefore, we did not adopt it.

3.4 CRFs

In CRFs, the probability of a particular label sequence y given observation sequence x is assigned as a normalized product of potential functions [13].

$$p(y \mid x, \lambda) = \frac{1}{Z(x)} \exp\left(\sum_j \lambda_j F_j(y, x) \right)$$

In above equation, $Z(x)$ is a normalization factor; λ_j are parameters to be estimated from training data; and

$$F_j(y, x) = \sum_{i=1}^{n} f_j(y_{i-1}, y_i, x, i)$$

where $f_j(y_{i-1}, y_i, x, i)$ is either a state function $s_j(y_i, x, i)$ of the label at position i and the observation sequence, or a transition function $t_j(y_{i-1}, y_i, x, i)$ of the entire observation and the labels at position i and position i - 1 in the label sequence.

Although we distinguished the patent and paper data, we used the same feature functions to train the CRFs model. Since only state functions were used, the difference among all functions is pertaining to the observation sequence. These observation sequences were defined as follows:

1. n-gram in the original sequence

2. n-gram in the POS-tag sequence

3. current POS-tag with other observed unigram and its POS-tag

The maximum size of n-gram is five. When unigram is adapted, the maximum distance from the observed unigram to current state is four. In other words, if the observed unigram is too far away from current state, then it was not considered in our CRFs model.

4. TAG MODIFICATION

The problem the previous model encountered is very low recall. A direct solution is to improve recall by increasing positive tags.

From CRFs, the $p(y \mid x, \lambda)$ is known. In other word, the probability of each state given the observation sequence could be calculated.

If the $p(Y = \text{"}other\text{"} \mid x, \lambda)$ i.e. the probability of the state recognized as "other" is not high enough, the "other" tag is modified by a positive tag. A positive tag is chosen as the replacement when its probability is the maximum among that of all positive tags.

So the update rule is as follows:

IF

$$p(Y = \text{"}other\text{"} \mid x, \lambda) < t \quad \text{// } t \text{ is a threshold}$$

THEN

$$p(y \mid x, \lambda) = \max_{Y \neq \text{"}other\text{"}} p(Y \mid x, \lambda)$$

$$y := \arg\max_{Y \neq \text{"}other\text{"}} p(Y \mid x, \lambda)$$

The basic idea is that we modified the negative tag if the model does not has enough confidence (here the threshold *t* was assigned as 90%) to a positive tag. The assigned positive tag has the highest confidence among all positive tags.

5. PATTERN-BASED METHOD

So far, the constructed model does not have the capability to solve two problems. First, the length of the observation sequence is too long. In this case, some indicator tokens, which are too far away from current state, are not involved in the model. This situation is very common in patents, because the sentence in patent is usually very long due to the use of preposition phrase or parallel structure.

The second problem is ambiguity. On one hand, it is difficult to differentiate ATTRIBUTE from TECHNOLOGY. The CRFs model only contains raw text and part-of-speech information, while both ATTRIBUTE and TECHNOLOGY is usually a noun phrase. Therefore, without additional knowledge, it is difficult to make a judgment whether a noun phrase is a TECHNOLOGY or an ATTRIBUTE. On the other hand, to differentiate VALUE like "reduce" from other verbs is also very difficult.

So we involve human knowledge and pattern-based method to address above two challenges. For the convenience of implementation, several text mining techniques were used. Thus, the final output results were the combination of CRFs-based method and pattern-based method.

5.1 Indicator Words for VALUE

The inspiration to design patterns is from the fact that VALUE and ATTRIBUTE are usually appearing in a pair, and moreover VALUE is either an adjective related to polarity opinion, namely good or bad, or a verb related to making some changes, for example, "improve", "facilitate", "adjust", "reduce" and "prevent".

Using the training data, we built a word list to cover all such indicator words and added more words into the list according to the semantics. We had taken the benefits of the WordNet, which is a thesaurus.

Next, the ATTRIBUTE is usually the nearest noun phrase to the VALUE. For example, if the VALUE equals to "improve", "improves", "improving" and "improvement", then the ATTRIBUTE is the nearest noun phrase after the VALUE; if the VALUE equals to "improved", then the ATTRIBUTE is the nearest noun phrase before the VALUE.

Therefore, the second step is searching noun phrases the indicator words are related to. We had tried two approaches.

5.2 Dependency Parsing - a Failed Trial

We firstly tried to utilize the Dependency Parsing to grasp the relation between VALUE and ATTRIBUTE. However, from the experiments, we realized that it is not a good idea.

Firstly, compared to POS-tagging, the Dependency Parsing does not offer more information; secondly, the dependency discovered cannot be used to link TECHNOLOGY, VALUE, and ATTRIBUTE; thirdly, many sentences in patent are too long or too difficult to parse; and finally, the parsing time is too long, for example, more than one hour is need to parse all sentences of patent topics in the dry run.

5.3 Chunking, Stopword and Laplacian

Since the Dependency Paring was failed to reap the relational entities, we designed a POS-based chunker to delimit noun-structure, and calculated the distance in the token sequence.

A noun-structure is a combination of sequential tokens in the original sequence. Compared to noun phrase, noun-structure is a broader concept. In other words, a noun phrase belongs to a noun-structure, but not all noun-structures are noun phrases. The reason why we used the noun-structure instead of noun phrase is because many ATTRIBUTE have a more complex structure than a noun phrase.

A fact must be taken aware of is that not all nearest noun-structure is ATTRIBUTE. Generally speaking, we got a rule: if the noun-structure contains one word, which can construct a common used pair with the corresponded indicator word, then the noun-structure cannot be accepted as an ATTRIBUTE. However, some common used pairs do not obey this rule. We noted that human can make the judgment based on semantics.

Therefore, a stopword list is built for every indicator word. The stopword list should cover as many as possible those words that obey above rule.

To build such a stopword list manually is very difficult. So it was learned from training data and the criterion is Laplacian:

$$Laplacian = \frac{e+1}{c+e+1}$$

where c is the number of correctly matched ATTRIBUTE and e be the number of errors. If the Laplacian is small than 0.5, then the pattern is accepted.

6. EVALUATION

We used the formal run's evaluation results released by the NTCIR-8 organizers to evaluate the performance of our system. In the formal run, the training data consists of 300 patent topics and 300 paper topics, while the test data is composed of 200 patent topics and 200 paper topics. The distribution of the desired entities is shown in Table 1.

Table 1. Distribution of the desired entities

Entity Type	Patent	Paper
Technology entities in title (TT)	39	93
Technology entities in abstract (AT)	847	342
Attribute entities in abstract (AA)	213	204
Value entities in abstract (AV)	198	193

We totally submitted three system runs: NUSME-1, NUSME-2 and NUSME-3, which are corresponding to output 1, output 2 and output 3 in Figure 1. The NUSME-1 adopted the CRFs method. Compared to the NUSME-1, the NUSME-2 added a tag modification step. The NUSME-3 enhanced the NUSME-2 by integrating the output of the pattern-based method. Therefore, from the first system run to the last system run, more and more efforts were involved.

In Figure 2 and Figure 3, the F-measure of one system is denoted by a bar; moreover the bar filled with sparse lines, dense lines, and black color denotes the F-measure of NUSME-1, NUSME-2, and NUSME-3, respectively. Figure 2 used the patent topics as input data, while the Figure 3 was created using the paper topics as input data.

As shown in Figure 2 and Figure 3, our second system run and third system run achieved relatively good results with respect to F-measure for both patent topics and paper topics. Specially, the last system run was the best among all participated system runs not only for patent topics but also for paper topics. It was as expected that more efforts obtained better results. A big improvement was achieved by the tag modification step.

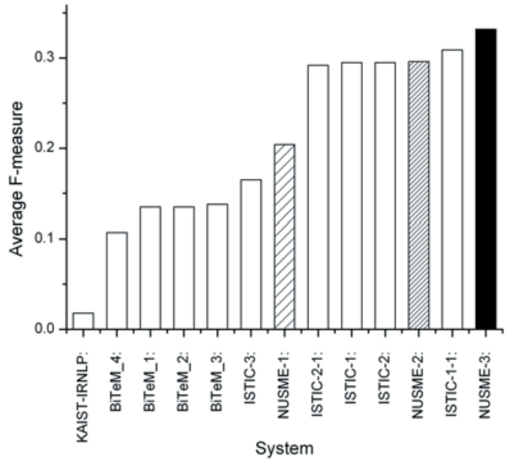

Figure 2. F-measure of all systems using patent topics

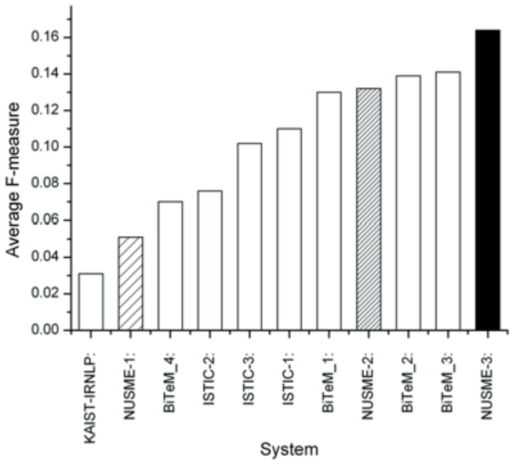

Figure 3. F-measure of all systems using paper topics

Figure 4 and Figure 5 shows the differences among three system runs using the patent data in details. The tag modification step, namely from NUSME-1 to NUSME-2, is able to improve the recall, because it enforce the CRFs model output more positive tags, therefore the chance of finding correct entities is increased. However, at the same time, additional output also has a high chance of reducing the precision. That is why precision of TT, AT and AV is reduced.

The increase precision of AA from NUSME-1 to NUSME-2 in Figure 5 is due to the very bad precision of AA in the first system run. Actually, there is no correct entity discovered in NUSME-1, so the precision of AA is zero. Therefore, once one correct entity is discovered in the second run, the precision of AA could be improved.

It can be observed that the manually designed patterns had improved both recall and precision of AA and AV. Because such patterns are designed to enhance the weakness of built CRFs model, and usually human intelligence is more accurate. There is no difference on TT and AT, because the patterns adopted are all related to attribute and value, not technology.

The CRFs method we adopted treated equally the TT, AT, AA and AV. However, TT and AT are quite different from AA and AV, because AA and AV, as discussed above, are relational entities i.e. they usually appear together. Such important fact was not considered in our CRFs method. As a supplement, the pattern-based method was designed by utilizing the relations between AA and AV. Therefore, the combination of both methods produced the best results.

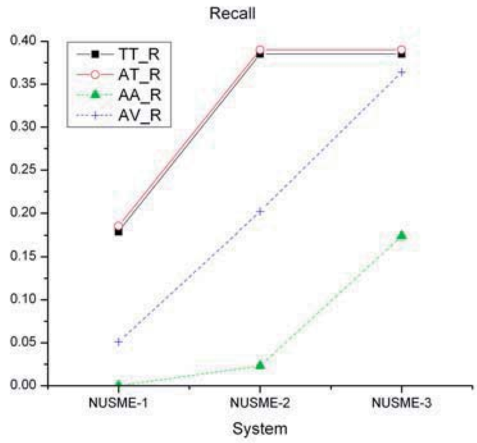

Figure 4. Recall of our system runs using patent data

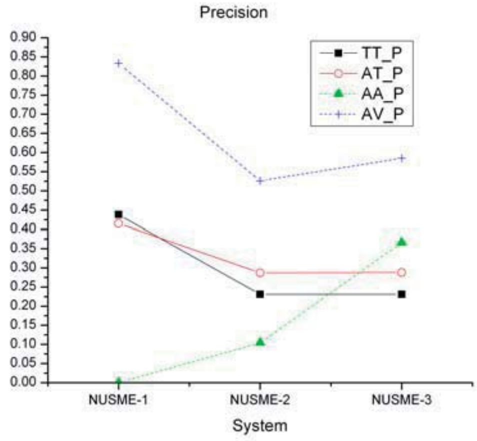

Figure 5. Precision of our system runs using patent data

The results of paper data, which can be observed from Figure 6 and Figure 7, are almost the same as that of patent data. The phenomenon demonstrates the consensus on the definition of desired entities. In other word, such definition is keeping the same with the changing of the documentation styles.

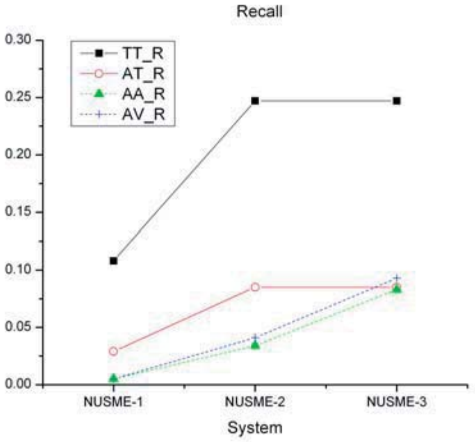

Figure 6. Recall of our system runs using paper data

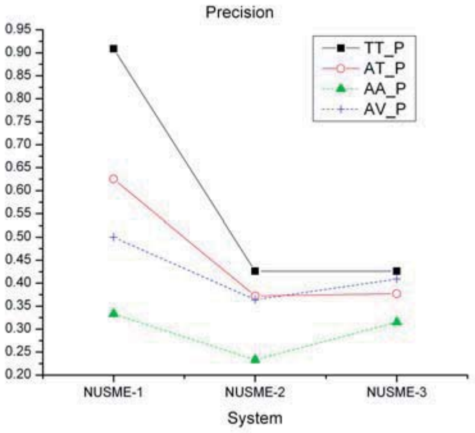

Figure 7. Precision of our system runs using paper data

7. CONCLUSIONS

In NTCIR-8 patent mining task, we built a system which adopted both statistical method and pattern-based method. Using our tag update rule is able to achieve a better F-measure than that of using the original CRFs method. Moreover, the pattern-based method we adopted seems making up for the weakness of using statistical method only. We achieved a relatively good result compared to other participants. Since the performance is still not good, we will further improve it in the future.

8. REFERENCES

[1] Fujii, A., Iwayama, M. and Kando, N. 2004. Overview of Patent Retrieval Task at NTCIR-4. In *Proceedings of the NTCIR-4* (Tokyo, June 2-4, 2004).

[2] SHINMORI, A. and OKUMURA, M. 2004. Can Claim Analysis Contribute toward Patent Map Generation ? In *Proceedings of the NTCIR-4* (Tokyo, June 2-4, 2004).

[3] UCHIDA, H. and MANO, A. 2004. Patent Map Generation using Concept-based Vector Space Model. In *Proceedings of the NTCIR-4* (Tokyo, June 2-4, 2004).

[4] Tseng, Y.-H., Lin, C.-J. and Lin, Y.-I. 2007. Text mining techniques for patent analysis. *Information Processing and Management*. 43 (2007), 1216-1247.

[5] Sarawagi, S. 2007. Information Extraction. *Foundations and Trends in Databases*. 1, 3 (2007), 261-377.

[6] Soderland, S. 1999. Learning Information Extraction Rules for Semi-Structured and Free Text. *Machine Learning*. 34, 1-3 (1999).

[7] Xiao, J., Chua, T.-S. and Liu, J. 2003. A Global Rule Induction Approach to Information Extraction. In *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence* (2003).

[8] Lafferty, J., McCallum, A. and Pereira, F. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the ICML* (2001).

[9] Sha, F. and Pereira, F. 2003. Shallow Parsing with Conditional Random Fields. In *Proceedings of the HLT/NAACL* (2003).

[10] Rosendfeld, B., Feldman, R. and Fresko, M. 2006. TEG - a hybrid approach to information extraction. *Knowledge Information Systems*. 9 (2006), 1-18.

[11] Toutanova, K., Klein, D., Manning, C. and Singer, Y. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of the HLT-NAACL 2003* (2003).

[12] Toutanova, K. and Manning, C. D. 2000. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. In *Proceedings of the Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)* (Hong Kong, 2000).

[13] Wallach, H. M. 2004. *Conditional Random Fields: An Introduction*. Technical Report. University of Pennsylvania, 2004.