Semi-Supervised Technical Term Tagging With Minimal User Feedback

Behrang QasemiZadeh [#], Paul Buitelaar [#], Tianqi Chen [*], Georgeta Bordea [#]

[#] Unit for Natural Language Processing, DERI, National University of Ireland, Galway
[*] Apex Data & Knowledge Management Lab, Shanghai Jiao Tong University
[#] firstname.lastname@deri.org, [*] tqchen@apex.sjtu.edu.cn

Abstract

In this paper, we address the problem of extracting technical terms automatically from an unannotated corpus. We introduce a technology term tagger , that is based on Liblinear Support Vector Machines and employs linguistic features including Part of Speech tags and Dependency Structures, in addition to user feedback to perform the task of identification of technology related terms. Our experiments show the applicability of our approach as witnessed by acceptable results on precision and recall.

Keywords: Technical Term Tagging, Information Extraction, Machine Learning

1. Introduction

The task of Technology Structure Mining is concerned with extracting information about technologies and their interdependencies in a given corpus of scientific literature (QasemiZadeh, 2010). An initial step towards fulfilling Technology Structure Mining is the identification of technology related terms (TRT). In this paper, we show that it is possible to build a language model for automatic extraction of TRT from an unannotated corpus by limited use of user feed-back.

We describe an approach for learning patterns to extract TRT. In short, for a given corpus of scientific literature, we first search for the terms collocated with seed words such as "technology" following similar research on the use of seed words in word sense disambiguation (Yarowsky, 1992) and semantic lexicon construction (Riloff and Shepherd, 1997). The extracted terms are then shown to the user who manually labels the extracted terms as a valid or invalid TRT. Our experiments over different corpora have shown that the number of such collocations is usually very low relative to the size of corpus, e.g., around 250 collocations for a corpus of size 3.5m tokens, and thus these collocations can be annotated in a reasonable time by an expert. In the next step, a dataset based on the validated TRTs is created automatically, where each individual occurrence of a TRT in a sentence is considered to be a record in the dataset. This dataset subsequently is used for training a SVM model for annotating additional TRTs in the corpus.[1]

The rest of the paper is organized as follows: in the next section we describe our methodology. In section 3., we introduce the datasets that are used for our experiments and the training phase. The experimental setup and the results are illustrated in section 4.. Related work is discussed in section 5.. Finally we conclude in section 6..

2. Learning from Unlabelled Corpora using Limited User Feedback

Our method builds upon a system comprising of a processing pipeline (Figure 1) and an entity relationship based data storage facility that provides us with a high level text query capability similar to the *"Corpus Query Language"*[2].

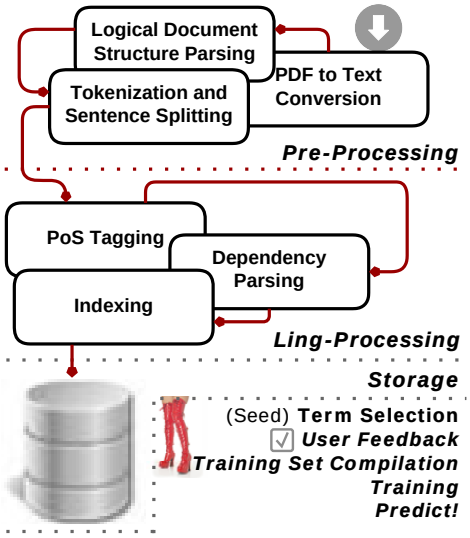

Figure 1: An Illustration of the Methodology

A pre-processing component provides facilities for extracting the text and further services for text segmentation; this is mainly based on ParsCit (Councill et al., 2008). Moreover, we use OpenNLP [3] for tokenization and sentence splitting.

The Stanford PoS tagger (Toutanova and Manning, 2000) is used for feeding PoS tagged sentences to the Malt Parser (Nivre, 2003) to get projective dependency parses of input sentences. The storage component performs indexing of the generated data in a way that each linguistically well defined unit can be identified uniquely alongside with its frequency over the corpus.

The post-indexing process involves building and using a dataset for the task of TRT identification with minimal de-

[1] The system is online available at `http://parsie.deri.ie/EEYORETTT`

[2] `http://www.fi.muni.cz/~thomas/corpora/CQL/`

[3] `http://incubator.apache.org/opennlp/index.html`

pendency on user feedback. Processes are as follows:

- Term Selection: Comprises a PoS-based heuristic that identifies collocations with the seed word "technology" in the corpus. An example output of this process over a corpus of publications in natural language processing is as follows:

  1. "Natural Language Processing"
  2. "Machine Translation"
  3. "Implementation"

- User Feedback: Selected terms are shown to the user who will be asked to identify positive examples from this list, e.g., the user may identify 1 and 2 above as positive examples.

- Training Set Compilation: All occurrences of selected terms in the previous step are retrieved from the corpus and each term occurrence is considered a positive or negative record in the training set, e.g., the following sentences with "machine translation" and "implementation" will be positive and negative records respectively:

  – "Around 1972 Colmerauer passed through the Stanford AI Lab, describing Prolog for the first time but, as you may or may not remember, as a tool for ***machine translation***" (Wilks, 2008)

  – "Its only when we get that first e-mail asking for the **implementation** of a method discussed in Computational Linguistics that the issue arises, and by then its too late." (Pedersen, 2008)

  The resulting dataset of positive and negative records and corresponding sentences is used for training.

- Training and Prediction: We employ Liblinear Support Vector Machine (Fan et al., 2008) for training and prediction purposes. We decided to use linear classification for two reasons: the ratio of the size of the feature space to the size of the training set (Lin et al., 2008) (see section 3.), and the fact that linear classification tends to be faster compared to SVM's kernel methods for applications such as the one proposed here (Cassel, 2009).

2.1. Feature Extraction

We use a set of binary valued features that characterize the term and its context. We used only the PoS sequence of the term as a feature for the term taken in isolation but we introduced several features for the context lexemes (i.e. the words surrounding the term):

- N-Grams on uniquely indexed lexemes (where n=1,2,3, 4) for the lexemes before and after the term

- N-Grams over PoS tags (where n=1,2,3, 4) for the lexemes before and after the term

- Grammatical relations to the lexemes in selected terms

2.2. Candidate Phrase Generation

At the predict phase we generate candidate terms similarly to the process of term selection at the training phase, however with relaxed conditions for PoS sequences. In effect, each permutation of words in a sentence is considered to be a candidate phrase unless the permutations contain words tagged with the following PoS tags: *VB, VBD, VBZ, WP, WDT, WRB, EX, JJS, LS, MD, PDT, UH, RB*. The candidate phrase also cannot start with PoS *V**.

3. Datasets and Training

We perform our experiments over two different corpora: Section A of the ACL Anthology Reference Corpus (ACL-ARC-A) (Bird et al., 2008), as well as the Semantic Web Dog Food corpus (SWDF) available from `http://data.semanticweb.org/`. The first corpus is used for developing an SVM model, while the latter has been used for the evaluation of the model.

We extract the text for 294 publications of ACL-ARC-A comprising 44241 sentences, 1109883 tokens, and 51281 types. The performed dependency parsing using the Malt Parser gave a total number of 1065199 dependency structures where 498079 of them were identical over lexemes, independent of their position. As for SWDF, we were able to extract text for 799 publications. This comprises 147802 sentences, 3599096 tokens, and 82550 types. The number of dependencies over the sentences were 3449126 where 1212525 were identical, ignoring the positions of lexemes.

We performed term selection on ACL-ARC-A using the word "technology". This resulted in 61 terms were 26 terms were accepted by an expert of the domain as valid technical terms and the rest were marked as negative example. Searching the corpus for the 61 different terms resulted in finding 8189 mentions of the terms. The Feature Extraction process resulted in 58934 identical features where the total number of extracted features for all the terms was 195924.

Worthwhile to mention that the L1-regularized L2-loss support vector classification performs slightly better than other type of solvers. The reported numbers here may not be valid for the evaluation purposes since a term may occur both in training and testing. Section 4.1. reports experimental results over the dataset where we make sure there is no overlap between train and test data sets.

4. Experiments

We have designed two experimental setups as follows:

- Dividing the training set from the ACL ARC corpus into two different sets where we can make sure that they have no overlapping terms in the sets: in this experiment we bound ourselves only to the terms that have been extracted from the term selection phase.

- Train and test across domains, i.e., the application of a model that was developed on ACL-ARC-A (computational linguistics domain) on SWDF (Semantic Web domain).

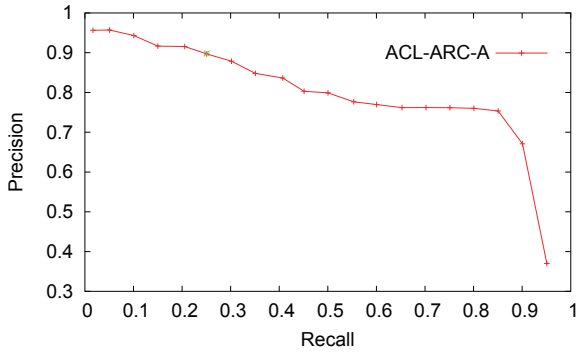

Figure 2: Precision-recall curve for 5-fold cross-validation

4.1. Evaluation on the Terms from the Term Selection Phase

In this experiment we limit ourselves to the terms selected and annotated by the user. For the ACL-ARC-A dataset, we randomly split instances in the training set into 5 folds, ensuring that each term occurs only in one fold. We have used a standard precision-recall curve over the output decision values from the classification algorithm to demonstrate the effectiveness of the trained model. Usually a higher precision at higher recall is treated as a better result. For each of the 5 folds in this experiment, we get the decision value from the model trained by the other folds. The decision values are then used to create the precision recall curve for evaluation as shown in figure 2. As the figure suggests, the trained model is stable and we can get a reasonable precision at about 30% recall. As there are no mutual terms between training and testing folds we can make sure that the curve is representative for the ability of the model for generalization.

4.2. Experimental Results over the Semantic Web Corpus

In the second experiment, we test our models over the Semantic Web Dog Food corpus. For each sentence in this corpus we generate candidate terms according to the method described in section 2.2.. After generating features for each candidate phrase we feed them into our binary classifiers. Applying the classifier based on the model for ACL-ARC-A resulted in 178360 identical terms labelled as positive.

As manual verification of all the terms is very time-consuming, we studied the behaviour of the classifier on the top-K results where K is 50, 100, 150, 200, 250 and 300. We make sure that the top-K terms are not introduced to the classifier during the training phase. We sort the list of classified terms based on the maximum decision value generated by the classifier for a classified term (Figure 3) as well as the sum of decision values for a term in the corpus (Figure 4). The latter ensures that we have considered the frequency of terms in the corpus.

5. Related Work

The lack of labelled corpora for term extraction has lead research in this area mostly in the direction of unsupervised approaches that rely either only on syntactic patterns

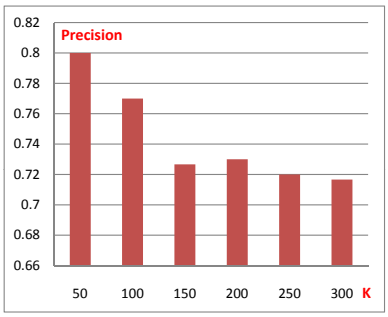

Figure 3: Top-K result sorted by Maximum Decision Value

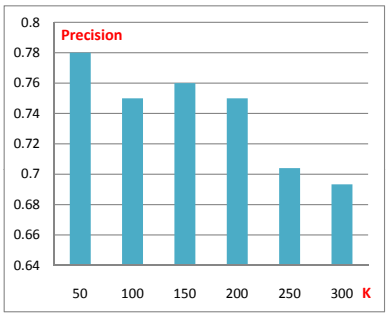

Figure 4: Top-K result sorted by the Sum of Decision Values

(Bourigault, 1996) or on a combination with statistical filters (Daille, 1996). We mitigate the lack of labelled corpora by manually annotating the technical terms that appear in the immediate vicinity of a few lexical triggers (i.e. "technology"). A similar work to the one proposed by us is reported in (Utiyama et al., 2000), where author-provided keywords are used for the training of a model for the extraction of technical terms.

Another related area is the identification of generalized names. (Yangarber et al., 2002) introduces the task of generalized names (GN) learning and compares this task with the task of Named Entity Recognition (Nadeau and Sekine, 2007) (NER) . In contrast to NER, GN recognition deals with the recognition of single- and multi-word domain-specific expressions and it can be more complex than proper names identification due to the absence of contextual cues such as word capitalization. Our work extends the task of GN recognition by identification of TRTs as class names (and class instances) in specific domains.

The work presented here intersects in many aspects with the work done in the area of Automatic Term Recognition (ATR). According to (Srajerova et al., 2009), ATR is the process of selecting elements in a corpus that are considered terms of the discipline which is the object of inquiry. A renewed interest in ATR has been reported especially because of its application in Ontology Learning and Population (Maynard et al., 2008).

In (Eichler, 2009), Eichler et al propose an unsupervised, domain-independent method for extraction of technical terms from scientific documents. In the proposed method, they first perform a nominal group chunking for extracting a list of candidate technical terms. Then, they classify these nominal groups into technical and non-technical terms us-

ing frequency counts retrieved from the MSN search engine. They evaluate their approach on three annotated corpora and report precision of 58% and recall of 81% in the best cases. Our approach relies on part of speech tags instead of nominal group chunkers and is therefore more easily extendable to other languages.

The problem of obtaining hand-labelled data from an unlabelled corpus is addressed by (Kozareva, 2006) in the context of the named entity recognition task. She proposes a system for extracting named entities that relies on automatically generated gazetteer lists that are used as features by the system. Her approach relies on prepositions as a contextual clue for extracting locations but in the case of technical terms tagging we showed that other categories of words are a more useful source of information.

A similar direction of research is proposed in (Saha et al., 2008), also in the context of the NE recognition task. The limitation of this approach is that the context of a word is restricted to a window of words of limited size. Our approach does not limit the number of contextual words analyzed for the extraction of patterns, putting restrictions only on the part of speech of each word.

6. Conclusion and Future Work

This paper investigates the use of a corpus-based machine learning approach for the task of technical term tagging. In the proposed method, we use collocations with word(s) such as "technology" for extracting seed technical terms. Subsequently, the extracted terms are classified by a user as technical and non-technical terms. Finally, a dataset based on the selected terms is created automatically and is used for training a SVM model to annotate additional technical terms in the corpus. Although we do not measure recall, our experimental results with respect to precision are quite favourable. More importantly, our work clearly shows that it is possible to generate a language model for the identification of technologically related terms, using semi-supervised learning with minimal user feedback.

Future work will focus on developing techniques for filtering and ranking the output of the proposed method. The application of an active learning scenario for improving the performance of the model based on current output is also under consideration.

7. Acknowledgements

This research is supported by Science Foundation Ireland grant SFI/08/CE/I1380(Lion-2) and Net2 Project funded by Marie Curie action Ïnternational Research Staff Exchange Scheme (IRSES), FP7-PEOPLE-2009-IRSES, under grant number 24761.

8. References

Steven Bird, Robert Dale, Bonnie J. Dorr, Bryan Gibson, Mark T. Joseph, Min yen Kan, Dongwon Lee, Brett Powley, Dragomir R. Radev, and Yee F. Tan. 2008. The ACL anthology reference corpus: A reference dataset for bibliographic research in computational linguistics.

D. Bourigault. 1996. Lexter: a natural language tool for terminology extraction. In *Proceedings of the 7th EURALEX International Congress*, pages 771–9.

Sofia Cassel. 2009. Maltparser and liblinear: Transition-based dependency parsing with linear classification for feature model optimization. Masters thesis in computational linguistics, Uppsala University, Department of Linguistics and Philology.

Isaac G Councill, C Lee Giles, and Min-Yen Kan. 2008. Parscit: An open-source crf reference string parsing package. *Proceedings of LREC*, (3):661–667.

B. Daille. 1996. Study and implementation of combined techniques for automatic extraction of terminology. In *The Balancing Act:Combining Symbolic and Statistical Approaches to Language*, pages 49–66. Cambridge, Mass.:MIT Press.

Hemsen H. Neumann G. Eichler, K. 2009. Unsupervised and domain-independent extraction of technical terms from scientific articles in digital libraries. In *Mandl, T., Frommholz, I. (eds.) Proc. of the Workshop "Information Retrieval", Organized as part of LWA*.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Zornitsa Kozareva. 2006. Bootstrapping named entity recognition with automatically generated gazetteer lists. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, EACL '06, pages 15–21, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chih-Jen Lin, R C Weng, and S S Keerthi. 2008. Trust region newton method for logistic regression. *The Journal of Machine Learning Research*, 9:627–650.

Diana Maynard, Yaoyong Li, and Wim Peters. 2008. Nlp techniques for term extraction and ontology population. In *Proceeding of the 2008 conference on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pages 107–127, Amsterdam, The Netherlands, The Netherlands. IOS Press.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, January.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT*, pages 149–160.

Ted Pedersen. 2008. Empiricism is not a matter of faith. *Comput. Linguist.*, 34:465–470, September.

Behrang QasemiZadeh. 2010. Towards technology structure mining from scientific literature. In *International Semantic Web Conference (2)*, pages 305–312.

Ellen Riloff and Jessica Shepherd. 1997. A corpus-based approach for building semantic lexicons. In *In Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 117–124.

Sujan Kumar Saha, Sudeshna Sarkar, and Pabitra Mitra. 2008. Gazetteer preparation for named entity recognition in indian languages. In *The 6th Workshop on Asian Languae Resources*.

Dominika Srajerova, Oleg Kovarik, and Vaclav Cvrcek.

2009. Automatic term recognition based on data-mining techniques. *Computer Science and Information Engineering, World Congress on*, 4:453–457.

Kristina Toutanova and Christopher D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *In EMNLP/VLC 2000*, pages 63–70.

Masao Utiyama, Masaki Murata, and Hitoshi Isahara. 2000. Using author keywords for automatic term recognition. *Terminology 6:2*, pages 313–326.

Yorick Wilks. 2008. On whose shoulders? *Comput. Linguist.*, 34:471–486, December.

Roman Yangarber, Winston Lin, and Ralph Grishman. 2002. Unsupervised learning of generalized names. In *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.

David Yarowsky. 1992. Word-sense disambiguation using statistical models of roget's categories trained on large corpora. In *Proceedings of the 14th conference on Computational linguistics - Volume 2*, COLING '92, pages 454–460, Stroudsburg, PA, USA. Association for Computational Linguistics.