Automating the Generation of Semantic Annotation Schema Using a Clustering Technique*

Vitór Souza[1], Nicola Zeni[1], Nadzeya Kiyavitskaya[1], Periklis Andritsos[1], Luisa Mich[2], and John Mylopoulos[1]

[1] Dept. of Information Engineering and Computer Science
[2] Dept. of Computer and Management Sciences,
University of Trento, Italy

Abstract. In order to generate semantic annotations for a collection of documents, one needs an annotation schema consisting of a semantic model (a.k.a. ontology) along with lists of linguistic indicators (keywords and patterns) for each concept in the ontology. The focus of this paper is the automatic generation of the linguistic indicators for a given semantic model and a corpus of documents. Our approach needs a small number of user-defined seeds and bootstraps itself by exploiting a novel clustering technique. The baseline for this work is the Cerno project [8] and the clustering algorithm LIMBO [2]. We also present results that compare the output of the clustering algorithm with linguistic indicators created manually for two case studies.

1 Introduction

Semantic annotation is commonly recognized as the one of the cornerstones of the Semantic Web. To generate domain-dependent metadata a semantic annotation system utilizes a semantic model, a.k.a. ontology along with sets of linguistic indicators (keywords and patterns, usually constructed manually by a domain expert) that determine what text fragments are to be annotated. This work was conducted in the context of the Cerno [8] project to explore the applicability of some of the main ingredients of a supervised categorical clustering algorithm LIMBO [2] for producing linguistic indicators for a given semantic model. LIMBO was originally proposed for clustering structural information of database tuples in relational databases. The main motivation for applying this method is that it requires a limited amounts of training data to bootstrap the learning algorithm and of human intervention at the initial stage. Moreover, the distance measure employed in LIMBO works with categorical data (i.e. data that do not have an inherent order) unlike most clustering techniques. Our primary goal in this work is to verify whether such a lightweight approach can facilitate

* This work has been partially funded by the EU Commission through the SERENITY and WEE-NET projects and by Provincia Autonoma di Trento through the STAMPS project.

the construction of an annotation schema, given a semantic model and a training set of documents. The focus of this paper is automation of the generation of an annotation schema for a given semantic domain using a clustering approach. We evaluate the performance of the method on two different data sets and the related annotation schemas manually developed for the previous applications of Cerno. This paper is structured as follows. The baseline of the present work in sketched in Section 2. It introduces the semantic annotation framework Cerno and LIMBO, the clustering technique adopted. Section 3 describes how the baseline technologies were extended and shows the tool built on top of the LIMBO. Section 4 presents the setup and evaluation of two experimental case studies and summarizes the lessons learned. Section 5 recalls the related work. Finally, conclusions are drawn in Section 6.

2 Research Baseline

2.1 Cerno semantic annotation framework

Cerno is a lightweight semantic annotation framework that exploits fast and scalable techniques from the software reverse engineering area. To annotate input documents, Cerno uses context-free grammars, generates a parse tree, and applies transformation rules to generate output in a target format [8]. The reader can find a detailed description of the architecture and the performance of the system in [8]. Normally, adapting Cerno to a new application domain requires a couple of weeks, because its domain dependent components have to be tuned for a given type of documents and a specific semantic model. In this work we explore the possibilities to automate the generation of such indicators for specific semantic domain. Having a set of examples, one can try to identify a set of contextual keywords describing relevant concepts using well-established statistical methods that have been proven effective in many areas. To this end, we have been experimenting with a scalable hierarchical categorical clustering algorithm called LIMBO [2].

2.2 Data Clustering with LIMBO

Data clustering [7] is a common technique for statistical data analysis and is widely used in many fields. Our approach is based on LIMBO [2], a scalable hierarchical categorical clustering algorithm that builds on the Information Bottleneck (IB) [11] framework for quantifying the relevant information preserved when clustering. The algorithm proceeds in three phases: Phase 1 constructs a cluster representative for the initial data set for efficiency purposes, Phase 2 performs the clustering on the representative and Phase 3 labels the initial input with the appropriate cluster information.

In our work we assume that apart from the initial data set, we give the algorithm as input an initial clustering. This clustering corresponds to the set of input records that contain the keywords a user indicated as seeds for the underlying semantic domain. As a consequence, we process the data in a hierarchical

fashion, starting with the initial clustering of the documents and proceeding until all relevant text fragments have been identified.

The method proceeds as follows: (1) Given a clustering C, group the set of input fragments T into the corresponding clusters S_t; (2) Merge all fragments of S_t into a representative R_t; (3) Find the fragments of $S \backslash S_t$ that are closest to the representative R_t; (4) Analyze the fragments found in step 3 for new semantic annotations of the domain and add them to S_t; (5) Repeat from step 2 until stopping criteria are satisfied.

3 Generation of the Annotation Schema

To provide a user assistance in generating linguistic indicators for Cerno's semantic annotation process the LIMBO algorithm was integrated in a user-friendly tool. Having such a support will allow to quickly adapt the framework to new application domains in terms of both different annotation schemas and types of documents. This tool has a graphical user interface (GUI) developed in Java. The input to LIMBO includes: the initial data set that is then transformed into a cluster representative by the clustering algorithm and a set of documents which are used for training. The graphic interface provides a step-by-step wizard that allows the user to configure the experiment, then separates the input file into clusters, repeatedly runs the algorithm and provides the results of each run. To initialize LIMBO, on the first step of the wizard the user specifies the following input files and parameters:

- *The input document* is the original unannotated input document.
- *The clusters file* is the text file that contains the clustering information.
- *The stopping criterion* specifies the way of terminating the algorithm.
- *The parsing mode* defines how the clusters will be generated from the input document. In our case, we take n words starting at the 1^{st} word of the sentence, then n words starting at the 2^{nd} word, and so forth. Thus, the n-grams parsing mode generates the largest number of clusters and consequently requires longer processing times.
- *The analyzer* is the module responsible for extracting the keywords out of the input document. Currently, the prototype contains two standard analyzers for English language with and without stemming that normalize the input text.

The tool parses the input file and separates the clusters using the specified parsing mode. Then, for each concept, it marks the clusters that contain any of the keywords of the category and runs the algorithm as many times as specified in the stopping criterion. When ran a fixed number of times, the k nearest neighbors are marked at each run. When finished, the prototype shows for each concept the most relevant words in each run of the algorithm.

4 Experimental Case Studies

To verify the feasibility of the proposed approach, we applied the LIMBO-based tool on two different experiments. The stopping criterion for the clustering algorithm was set to 10 iterations with the addition of the 2 nearest-neighbors. We selected 10 as the number of iterations empirically, given that the output of the clustering algorithm remained almost unchanged after a number of iterations greater than 10. The tool was run with 8 different parsing configurations per each experiment, half of them with stemming and half without, varying the parsing mode: sentences, all punctuation marks, 3-grams and 7-grams. Number 3 for n-grams mode was chosen to account for commonly used word collocations, such as for instance "information system" or "health care cleaninghouse", and number 7 was defined as the highest upper bound for a possible number of words in collocations.

We evaluated the performance by comparing automated results to a *Gold model*, i.e. the list of indicators drawn manually by the experts, and calculating recall and precision quality measures [12].

The HIPAA experiment. In the past a Cerno adaptation to the text of the Health Insurance Portability and Accountability Act (HIPAA) was generated by manual analysis of the document [3], annotating document fragments describing rights, anti-rights, obligations, anti-obligations, and related constraints. Thus, the purpose of this experiment was to evaluate how many of these indicators can be extracted by the clustering technique. We used as input four semantic categories and several corresponding keyword-seeds. Overall, in this experiment the tool has demonstrated low recall (from 0.13 to 0.38) , except for the *Condition* concept (0.75). Better results were obtained for the runs with the stemming analyzer. Among the unstemmed results, the best average score is delivered by the 3-grams parsing mode. The processing times changes depending on parsing mode. In particular, n-grams mode causes generation of a larger number of clusters from the input document, compared to other two modes, thus increasing processing times of the algorithm (average 25.5 min against 3.75 min for non n-grams runs).

The accommodation ads experiment. In our previous work [9], to annotate advertisements for accommodation in Rome drawn from an on-line newspaper, we used the annotation schema which represented the information needs of a tourist and included the concepts: *Accommodation Type, Contact, Facility, Term (of availability), Location,* and *Price*. The lists with linguistic indicators were constructed by hand from a set of examples. This experiment utilized the same input documents and categories from an earlier experiment using accommodation ads retrieved from tourism websites. Selected randomly, one third of the keywords found through the manual extraction process performed previously were included in the clusters file. This experiment has shown results of higher quality in respect to both recall and precision values (many runs above 0.6).

The runs with the stemming option turned off have demonstrated higher scores. Either with or without stemming, the best average scores were obtained for the 3-grams parsing mode.

Discussion of results. The evaluation results suggest that it is most effective to use the 3-grams parsing mode, to obtain the output of the best quality either for stemmed or non-stemmed processing. 3-grams parsing mode generates the largest number of clusters from the input text, thus essentially increasing processing times of the LIMBO algorithm.

The legal documents turned out to be more difficult for automated generation of linguistic indicators. This shortcoming is caused by the nature of the concepts of interest. Right, obligation, condition, and exception are very abstract entities and normally span relatively large text fragments, which makes it difficult to apply clustering techniques to identify appropriate contextual keywords. While short ads documents written in a very precise language and having similar structure provide a better learning environment for the LIMBO method.

Although it may seem that the Limbo-based technique does not achieve desired high recall values, some keywords found do not appear in the hand-crafted list (and thus were not counted as true positives), but were found relevant by a human judge. Therefore, we believe that LIMBO can provide a more complete approach to populate annotation schema with domain-specific indicators. Results produced by LIMBO can be a good starting point for a human expert when working with a new semantic domain. Using clustering techniques we are better able to support the generation of new annotation schemas in a systematic way. To further improve the LIMBO-based tool, we plan to provide a better guidance to the user thtough the underlying process.

5 Related Work

There are several proposals for weakly supervised methods intended to populate an ontology, a task similar to the generation of linguistic indicators. One of these is the *Class-Example* method [10] that exploits lexico-syntactic features to learn a classification rule from a seed set of terms. In contrast, the *Class-Pattern* approach [6] relies on using a set of patterns that indicate the presence of certain relationships, such as "is-a". *Class-Word* technique [5] uses contextual features to extract features in which a concept occurs.

Among the systems that use statistical techniques for populating semantic models is Ontosophie [4]. The system is based on machine learning natural language processing techniques to learn extraction rules for the concepts of a given ontology combining a shallow parsing tool called Marmot and a conceptual dictionary induction system called Crystal. The OntoPop [1] methodology strives for documents annotation and ontology population under a unified framework. In addition, it adopts two other tools: Intelligent Topic Manager for representing and managing the domain model and Insight Discoverer Extractor for extracting information from texts.

6 Conclusions and Future Work

In this work, we explore the problem of generating linguistic indicators for semantic annotation tools. The contribution of this paper consists of utilizing novel statistical clustering techniques and in particular is inspired by LIMBO [2] in order to automatically generate these indicators. Moreover, in order to allow experimenting with clustering techniques and facilitate the user's work, a tool implementing the LIMBO algorithm was developed in Java. We verified the effectiveness of the proposed technique in two different case studies.

Our future work includes further experimentation with different configurations of the clustering technique in order to improve the quality of results produced. As well, we propose to actually run semantic annotation experiments using the linguistic indicators generated by the LIMBO tool, in order to better assess their effectiveness.

References

1. Amardeilh, F.: OntoPop or how to annotate documents and populate ontologies from texts. In: Proc. of the ESWC 2006 Workshop on Mastering the Gap: From Information Extraction to Semantic Representation, Budva, Montenegro (2006)
2. Andritsos, P., Tsaparas, P., Miller, R. J., Sevcik, K. C.: LIMBO: Scalable Clustering of Categorical Data. In: Proc. of EDBT'04 (2004)
3. Breaux, T. D., Vail, M. W., Antón, A. I.: Towards regulatory compliance: Extracting rights and obligations to align requirements with regulations. In: Proc. of RE'06, pp. 46–55, Washington, DC, USA, IEEE Computer Society (2006)
4. Celjuska, D., Vargas-Vera, M.: Ontosophie: A Semi-Automatic System for Ontology Population from Text. In: Proc. of ICON'04, Hyderabad, India (2004)
5. Cimiano, P., Völker, J.: Towards Large-Scale, Open-Domain and Ontology-Based Named Entity Classification. In: Proceedings of RANLP'05, pp. 166–172 (2005)
6. Hearst, M.: Automated Discovery of WordNet Relations. In: WordNet: An Electronic Lexical Database, Christiane Fellbaum (ed.) MIT Press (1998)
7. Jardine, N., Sibson, R.: The construction of hierarchic and non-hierarchic classifications. The Computer Journal, vol. 11, pp. 117–184 (1968)
8. Kiyavitskaya, N., Zeni, N., Mich, L., Cordy, J. R., Mylopoulos, J.: Text mining through semi automatic semantic annotation. In: Proc. of PAKM'06. LNCS, vol. 4333, pp. 143-154. Springer-Verlag (2006)
9. Kiyavitskaya, N., Zeni, N., Mich, L., Cordy, J. R., Mylopoulos, J.: Annotating Accommodation Advertisements using CERNO. In: Proc. of ENTER'07, pp. 389–400. Springer Verlag, Wien (2007)
10. Tanev, H., Magnini, B.: Weakly Supervised Approaches for Ontology Population. In: Proc. of EACL'06, Trento, Italy (2006)
11. Tishby, N., Pereira, F.C., Bialek, W.: The Information Bottleneck Method. In 37th Annual Allerton Conf. on Communication, Control and Computing (1999)
12. Baeza-Yates R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison Wesley (1999)