The GOD model

Alfio Massimiliano Gliozzo
ITC-irst
Trento, Italy
gliozzo@itc.it

Abstract

GOD (General Ontology Discovery) is an unsupervised system to extract semantic relations among domain specific entities and concepts from texts. Operationally, it acts as a search engine returning a set of true predicates regarding the query instead of the usual ranked list of relevant documents. Our approach relies on two basic assumptions: (i) paradigmatic relations can be established only among terms in the same Semantic Domain an (ii) they can be inferred from texts by analyzing the Subject-Verb-Object patterns where two domain specific terms co-occur. A qualitative analysis of the system output shows that GOD provide true, informative and meaningful relations in a very efficient way.

1 Introduction

GOD (General Ontology Discovery) is an unsupervised system to extract semantic relations among domain specific entities and concepts from texts. Operationally, it acts as a search engine returning a set of true predicates regarding the query instead of the usual ranked list of relevant documents. Such predicates can be perceived as a set of semantic relations explaining the domain of the query, i.e. a set of binary predicated involving domain specific entities and concepts. Entities and concepts are referred to by domain specific terms, and the relations among them are expressed by the verbs of which they are arguments.

To illustrate the functionality of the system, below we report an example for the query *God*.

```
god:
lord hear prayer
god is creator
god have mercy
faith reverences god
lord have mercy
jesus_christ is god
god banishing him
god commanded israelites
god was trinity
abraham believed god
god requires abraham
god supply human_need
god is holy
noah obeyed god
```

From a different perspective, GOD is first of all a general system for ontology learning from texts (Buitelaar et al., 2005). Likewise current state-of-the-art methodologies for non-hierarchical relation extraction it exploits shallow parsing techniques to identify syntactic patterns involving domain specific entities (Reinberger et al., 2004), and statistical association measures to detect relevant relations (Ciaramita et al., 2005). In contrast to them, it does not require any domain specific collection of texts, allowing the user to describe the domain of interest by simply typing short queries. This feature is of great advantage from a practical point of view: it is obviously more easy to formulate short queries than to collect huge amounts of domain specific texts.

Even if, in principle, an ontology is supposed to represent a domain by a hierarchy of concepts and entities, in this paper we concentrate only on the non-hyrarchical relation extraction process. In addition, in this work we do not address the problem of associating synonyms to the same concept (e.g. *god* and *lord* in the example above).

In this paper we just concentrate on describing our general framework for ontology learning, postponing the solution of the already mentioned problems. The good quality of the results and the well foundedness of the GOD framework motivate our future work.

2 The GOD algorithm

The basic assumption of the GOD model is that paradigmatic relations can be established only among terms in the same Semantic Domain, while concepts belonging to different fields are mainly unrelated (Gliozzo, 2005). Such relations can be identified by considering Subject-Verb-Object (SVO) patterns involving domain specific terms (i.e. syntagmatic relations).

When a query $Q = (q_1, q_2, \ldots, q_n)$ is formulated, GOD operates as follows:

Domain Discovery Retrieve the ranked list $dom(Q) = (t_1, t_2, \ldots, t_k)$ of domain specific terms such that $sim(t_i, Q) > \theta'$, where $sim(Q, t)$ is a similarity function capturing domain proximity and θ' is the *domain specificity* threshold.

Relation Extraction For each SVO pattern involving two different terms $t_i \in dom(Q)$ and $t_j \in dom(Q)$ such that the term t_i occurs in the subject position and the term t_j occurs in the object position return the relation $t_i v t_j$ if $score(t_i, v, t_j) > \theta''$, where $score(t_i, v, t_j)$ measures the syntagmatic association among t_i, v and t_j.

In Subsection 2.1 we describe into details the Domain Discovery step. Subsection 2.2 is about the relation extraction step.

2.1 Domain Discovery

Semantic Domains (Magnini et al., 2002) are clusters of very closely related concepts, lexicalized by domain specific terms. Word senses are determined and delimited only by the meanings of other words in the same domain. Words belonging to a limited number of domains are called domain words. Domain words can be disambiguated by simply identifying the domain of the text.

As a consequence, concepts belonging to different domains are basically unrelated. This observation is crucial from a methodological point of view, allowing us to perform a large scale structural analysis of the whole lexicon of a language, otherwise computationally infeasible. In fact, restricting the attention to a particular domain is a way to reduce the complexity of the overall relation extraction task, that is evidently quadratic in the number of terms.

Domain information can be expressed by exploiting Domain Models (DMs) (Gliozzo et al., 2005). A DM is represented by a $k \times k'$ rectangular matrix $\mathbf{D}$, containing the domain relevance for each term with respect to each domain, where k is the cardinality of the vocabulary, and k' is the size of the Domain Set.

DMs can be acquired from texts in a totally unsupervised way by exploiting a lexical coherence assumption (Gliozzo, 2005). To this aim, term clustering algorithms can be adopted: each cluster represents a Semantic Domain. The degree of association among terms and clusters, estimated by the learning algorithm, provides a domain relevance function. For our experiments we adopted a clustering strategy based on Latent Semantic Analysis, following the methodology described in (Gliozzo, 2005). This operation is done off-line, and can be efficiently performed on large corpora. To filter out noise, we considered only those terms having a frequency higher than 5 in the corpus.

Once a DM has been defined by the matrix $\mathbf{D}$, the Domain Space is a k' dimensional space, in which both texts and terms are associated to Domain Vectors (DVs), i.e. vectors representing their domain relevances with respect to each domain. The DV $\vec{t_i'}$ for the term $t_i \in \mathcal{V}$ is the i^{th} row of $\mathbf{D}$, where $\mathcal{V} = \{t_1, t_2, \ldots, t_k\}$ is the vocabulary of the corpus. The similarity among DVs in the Domain Space is estimated by means of the cosine operation.

When a query $Q = (q_1, q_2, \ldots, q_n)$ is formulated, its DV $\vec{Q'}$ is estimated by

$$\vec{Q'} = \sum_{j=1}^{n} \vec{q_j'} \tag{1}$$

and then compared to the DVs of each term $t_i \in \mathcal{V}$ by adopting the cosine similarity metric

$$sim(t_i, Q) = \cos(\vec{t_i'}, \vec{Q'}) \tag{2}$$

where $\vec{t_i'}$ and $\vec{q_j'}$ are the DVs for the terms t_i and q_j, respectively.

All those terms whose similarity with the query is above the *domain specificity* threshold θ' are

then returned as an output of the function $dom(Q)$. Empirically, we fixed this threshold to 0.5. In general, the higher the domain specificity threshold, the higher the relevance of the discovered relations for the query (see Section 3), increasing accuracy while reducing recall. In the previous example, $dom(god)$ returns the terms *lord*, *prayer*, *creator* and *mercy*, among the others.

2.2 Relation extraction

As a second step, the system analyzes all the syntagmatic relations involving the retrieved entities. To this aim, as an off-line learning step, the system acquires Subject-Verb-Object (SVO) patterns from the training corpus by using regular expressions on the output of a shallow parser.

In particular, GOD extracts the relations $t_i v t_j$ for each ordered couple of domain specific terms (t_i, t_j) such that $t_i \in dom(Q)$, $t_j \in dom(Q)$ and $score(t_i, v, t_j) > \theta''$. The confidence score is estimated by adopting the heuristic confidence measure described in (Reinberger et al., 2004), reported below:

$$score(t_i, v, t_j) = \frac{\frac{F(t_i,v,t_j)}{\min(F(t_i),F(t_j))}}{\frac{F(t_i,v)}{F(t_i)} + \frac{F(v,t_j)}{F(t_j)}} \qquad (3)$$

where $F(t)$ is the frequency of the term t in the corpus, $F(t, v)$ is the frequency of the SV pattern involving both t and v, $F(v, t)$ is the frequency of the VO pattern involving both v and t, and $F(t_i, v, t_j)$ is the frequency of the SVO pattern involving t_i, v and t_j. In general, augmenting θ'' is a way to filter out noisy relations, while decreasing recall.

It is important to remark here that all the extracted predicates occur at least once in the corpus, then they have been *asserted* somewhere. Even if it is not a sufficient condition to guarantee their truth, it is reasonable to assume that most of the sentences in texts express true assertions.

The relation extraction process is performed online for each query, then efficiency is a crucial requirement in this phase. It would be preferable to avoid an extensive search of the required SVO patterns, because the number of sentences in the corpus is huge. To solve this problem we adopted an *inverted relation index*, consisting of three hash tables: the SV(VO) table report, for each term, the frequency of the SV(VO) patterns where it occurs as a subject(object); the SVO table reports, for each ordered couple of terms in the corpus, the frequency of the SVO patterns in which they co-occur. All the information required to estimate Formula 3 can then be accessed in a time proportional to the frequencies of the involved terms. In general, domain specific terms are not very frequent in a generic corpus, allowing a fast computation in most of the cases.

3 Evaluation

Performing a rigorous evaluation of an ontology learning process is not an easy task (Buitelaar et al., 2005) and it is outside the goals of this paper. Due to time constraints, we did not performed a quantitative and objective evaluation of our system. In Subsection 3.1 we describe the data and the NLP tools adopted by the system. In Subsection 3.2 we comment some example of the system output, providing a qualitative analysis of the results after having proposed some evaluation guidelines. Finally, in Subsection 3.3 we discuss issues related to the recall of the system.

3.1 Experimental Settings

To expect high coverage, the system would be trained on WEB scale corpora. On the other hand, the analysis of very large corpora needs efficient preprocessing tools and optimized memory allocation strategies. For the experiments reported in this paper we adopted the British National Corpus (BNC-Consortium, 2000), and we parsed each sentence by exploiting a shallow parser on the output of which we detected SVO patterns by means of regular expressions[1].

3.2 Accuracy

Once a query has been formulated, and a set of relations has been extracted, it is not clear how to evaluate the quality of the results. The first four columns of the example below show the evaluation we did for the query *Karl Marx*.

```
Karl Marx:
TRIM economic_organisation determines superstructure
TRUM capitalism needs capitalists
FRIM proletariat overthrow bourgeoisie
TRIM marx understood capitalism
???E marx later marxists
TRIM labour_power be production
TRIM societies are class_societies
?RIM private_property equals exploitation
TRIM primitive_societies were classless
TRIM social_relationships form economic_basis
TRIM max_weber criticised marxist_view
```

[1]For the experiments reported in this paper we used a memory-based shallow parser developed at CNTS Antwerp and ILK Tilburg (Daelemans et al., 1999) together with a set of scripts to extract SVO patterns (Reinberger et al., 2004) kindly put at our disposal by the authors.

```
TRIM contradictions legitimizes class_structure
?R?E societies is political_level
?R?E class_society where false_consciousness
?RUE social_system containing such_contradictions
TRIM human_societies organizing production
```

Several aspects are addressed: truthfulness (i.e. True vs. False in the first column), relevance for the query (i.e. Relevant vs. Not-relevant in the second column), information content (i.e. Informative vs. Uninformative, third column) and meaningfulness (i.e. Meaningful vs. Error, fourth column). For most of the test queries, the majority of the retrieved predicates were true, relevant, informative and meaningful, confirming the quality of the acquired DM and the validity of the relation extraction technique[2].

From the BNC, GOD was able to extract good quality information for many different queries in very different domains, as for example *music*, *unix*, *painting* and many others.

3.3 Recall

An interesting aspect of the behavior of the system is that if the domain of the query is not well represented in the corpus, the domain discovery step retrieves few domain specific terms. As a consequece, just few relations (and sometimes no relations) have been retrieved for most of our test queries. An analysis of such cases showed that the low recall was mainly due to the low coverage of the BNC corpus. We believe that this problem can be avoided by training the system on larger scale corpora (e.g. from the Web).

4 Conclusion and future work

In this paper we reported the preliminary results we obtained from the development of GOD, a system that dynamically acquires ontologies from texts. In the GOD model, the required domain is formulated by typing short queries in an Information Retrieval style. The system is efficient and accurate, even if the small size of the corpus prevented us from acquiring domain ontologies for many queries. For the future, we plan to evaluate the system in a more rigorous way, by contrasting its output to hand made reference ontologies for different domains. To improve the coverage of the system, we are going to train it on WEB scale text collections and to explore the use of supervised relation extraction techniques. In addition, we are improving relation extraction by adopting a more sophisticated syntactic analisys (e.g. Sematic Role Labeling). Finally, we plan to explore the usefulness of the extracted relations into NLP systems for Question Answering, Information Extraction and Semantic Entailment.

[2]It is worthwhile to remark here that evaluation strongly depends on the point of view from which the query has been formulated. For example, the predicate *private_property equals exploitation* is true in the Marxist view, while it is obviously false with respect to the present economic system.

Acknowledgments

This work has been supported by the ONTOTEXT project, funded by the Autonomous Province of Trento under the FUP-2004 research program. Most of the experiments have been performed during my research stage at the University of Antwerp. Thanks to Walter Daelemans and Carlo Strapparava for useful suggestions and comments and to Marie-Laure Reinberger for having provided the SVO extraction scripts.

References

BNC-Consortium. 2000. British national corpus.

P. Buitelaar, P. Cimiano, and B. Magnini. 2005. *Ontology learning from texts: methods, evaluation and applications*. IOS Press.

M. Ciaramita, A. Gangemi, E. Ratsch, J. Saric, and I. Rojas. 2005. Unsupervised learning of semantic relations between concepts of a molecular biology ontology. In *In proceedings of IJCAI-05*, Edimburgh, Scotland.

W. Daelemans, S. Buchholz, and J. Veenstra. 1999. Memory-based shallow parsing. In *Proceedings of CoNLL-99*.

A. Gliozzo, C. Giuliano, and C. Strapparava. 2005. Domain kernels for word sense disambiguation. In *Proceedings of ACL-05*, pages 403–410, Ann Arbor, Michigan.

A. Gliozzo. 2005. *Semantic Domains in Computational Linguistics*. Ph.D. thesis, University of Trento.

B. Magnini, C. Strapparava, G. Pezzulo, and A. Gliozzo. 2002. The role of domain information in word sense disambiguation. *Natural Language Engineering*, 8(4):359–373.

M.L. Reinberger, P. Spyns, A. J. Pretorius, and W. Daelemans. 2004. Automatic initiation of an ontology. In *Proceedings of ODBase'04*, pages 600–617. Springer-Verlag.