An Approach to Automated Learning of Conceptual Graphs from Text

F. Rotella[1], S. Ferilli[1,2], and F. Leuzzi[1]

[1] Dipartimento di Informatica – Università di Bari
{fulvio.rotella, stefano.ferilli, fabio.leuzzi}@uniba.it
[2] Centro Interdipartimentale per la Logica e sue Applicazioni – Università di Bari

Abstract. Many document collections are private and accessible only by selected people. Especially in business realities, such collections need to be managed, and the use of an external taxonomic or ontological resource would be very useful. Unfortunately, very often domain-specific resources are not available, and the development of techniques that do not rely on external resources becomes essential. Automated learning of conceptual graphs from restricted collections needs to be robust with respect to missing or partial knowledge, that does not allow to extract a full conceptual graph and only provides sparse fragments thereof. This work proposes a way to deal with these problems applying relational clustering and generalization methods. While clustering collects similar concepts, generalization provides additional nodes that can bridge separate pieces of the graph while expressing it at a higher level of abstraction. In this process, considering relational information allows a broader perspective in the similarity assessment for clustering, and ensures more flexible and understandable descriptions of the generalized concepts. The final conceptual graph can be used for better analyzing and understanding the collection, and for performing some kind of reasoning on it.

1 Introduction

Many document collections are private and accessible only by selected people. Especially in business realities, such collections need to be managed in order to carry out tasks as retrieval and extraction of information. Unfortunately, obtaining automatically *Full Text Understanding* is not trivial, due to the intrinsic ambiguity of natural language and to the huge amount of common sense and linguistic/conceptual background knowledge needed to switch from a purely syntactic representation to the underlying semantics. Nevertheless, even small portions of such knowledge may significantly improve understanding performance, at least in limited domains. Although standard tools, techniques and representation formalisms are still missing, lexical and/or conceptual graphs[1] can provide a useful support to many NLP tasks, allowing automatic systems to exploit

[1] We refer to the term 'conceptual graph' as a synonym for 'concept network', with no reference to Sowa's formalism.

different kinds of relationships that are implicit in the text but required to correctly understand it. Although the use of an external taxonomic or ontological resource can be very useful for these purposes, very often domain-specific resources of this kind are not available, and manually building them is very costly and error-prone. This encourages research for techniques that can work without the need for these facilities, and possibly even automatically construct them by mining large amounts of documents in natural language. Extending a previous work, this paper proposes a technique to automatically extract conceptual graphs from text and reason with them without any external knowledge. Since automated learning of conceptual graphs from small collections needs to be robust with respect to missing or partial knowledge, we first apply a clustering phase to group similar concepts, and then generalize the items in each cluster to obtain new concepts that can be used to build taxonomic relations in the graph, in some cases bridging disjoint portions of the graph. More specifically, instead of using *flat* attribute-value data we adopt a relational perspective, that allows to compare objects with more informative descriptions than those provided by their shared attributes only (that might be very few in such context).

This work is organized as follows: the next section describes related works; Section 3 outlines the proposed approach; then we present an evaluation of our solution; lastly we conclude with some considerations and future works.

2 Related Work

Many approaches have been attempted to build conceptual graphs, taxonomies and ontologies from text. In general there are two main strategies: the former builds them exploiting only what is contained in the texts without external knowledge, and the latter exploits some external resources to do the same. The former approach is particularly indicated in such business realities where do not exist any kind of structured and *Machine-readable* external knowledge as a domain taxonomy/ontology. In this setting [1] builds concept hierarchies using Formal Concept Analysis by grouping objects with their attributes, which are determined from text linking terms with verbs. Conversely, our approach is focused on the building of a semantic network relying on the whole graph of concepts and relationships, rather than only on shared attributes as in their taxonomical representation of concepts. Another approach to ontology discovering is proposed in [11], in which the author defines a language to build formal ontologies by deductive discovery in similar way with logic programming. In particular, the author defines both a specific language for manipulating Web pages and a logic program to discover concept lattice. Conversely, we do not limit the kind of relationships to a predefined set, new relationships are created when a never seen verbal relationship between concepts is found. The latter approach differs to the ours, since it strongly depends on external knowledge for the building of taxonomies and/or ontologies. In particular [10, 9] build ontologies by labelling taxonomic relations only; [13] builds a taxonomy considering only concepts that are present in a domain but do not appear in others; [5] uses a combination of existing linguistic resources (VerbNet [6] and WordNet [3]) to shift the representation to the semantic level.

Regarding our proposal, for the syntactic analysis of the input text we exploit the *Stanford Parser* and *Stanford Dependencies* [7, 2], two tools that can identify the most likely syntactic structure of sentences (including active/passive and positive/negative forms), and specifically 'subject' or '(direct/indirect) object' components. They also normalize the words in the input text using lemmatization instead of stemming, which allows to distinguish their grammatical role and is more comfortable to read by humans.

Since the subject of the sentences is usually written only at its first occurrence and then it is replaced by pronouns, we face the anaphora resolution task exploiting JavaRAP that is an implementation of the classic Resolution of Anaphora Procedure [14]. It resolves third person pronouns, lexical anaphors, and identifies pleonastic pronouns.

We also exploited JUNG [12] (Java Universal Network/Graph Framework), which provides a common and extendible language for the modelling, analysis, and visualization of data that can be represented as a graph or network.

Lastly, we need to assess the similarity among concepts in a given conceptual graph. This has been achieved exploiting the similarity measure between Horn clauses proposed in [4]. This measure applies a layered evaluation that, starting from simpler components, proceeds towards higher-level ones repeatedly applying a basic similarity formula, and exploiting in each level the information coming from lower levels and extending it with new features. At the basic level are terms (i.e., constants or variables in a Datalog setting), that represent objects in the world and whose similarity is based on their properties (expressed by unary predicates) and roles (expressed by their position as arguments in n-ary predicates). The next level involves atoms built on n-ary predicates, whose similarity is based on their "star" (the multiset of predicates corresponding to atoms directly linked to them in the clause body, that expresses their similarity 'in breadth') and on the average similarity of their arguments. Then, the similarity of sequences of atoms is based on the length of their compatible initial subsequence and on the average similarity of the atoms appearing in such subsequence. Finally, the similarity of clause is computed according to their least general generalization, considering how many literals and terms they have in common and on their corresponding lower-level similarities.

3 Proposed Approach

This proposal relies on a previous work [8], assuming that each noun in the text corresponds to an underlying *concept* (phrases can be preliminarily extracted using suitable techniques, and handled as single terms). A concept is defined by the set of the others concepts that interact with it in the world described by the corpus. The outcome is a graph, where nodes are the concepts/nouns recognized in the text, and edges represent the relationships among these nodes, expressed by verbs in the text (the direction of edges denotes the role of the associated nodes in the relationship). In particular, for each sentence we keep into account also the positive or negative valence of each verbs.

3.1 Conceptual Graph Construction

In [8] the input text was directly processed to build a conceptual graph, discarding the relationships (verbs) associated to pronouns because they could not be associated to specific concepts. In order to face this lack in this work, we have pre-processed the input text using JavaRAP, replacing pronouns with the corresponding nouns. Then, the set of relationships of these nouns has been extended, improving the quality of the resulting conceptual graph. The obtained text is processed using the Stanford Parser, in order to extract the syntactic structure of the sentences that make it up. In particular, we are interested only in (active or passive) sentences of the form *subject-verb-(direct/indirect)complement*, from which we extract the corresponding triples $\langle$*subject*, *verb*, *complement*$\rangle$ that will provide the concepts (the *subject*s and *complement*s) and relations (*verb*s) for the graph. Furthermore, indirect complements are treated as direct ones, by embedding the corresponding preposition into the verb: e.g., *to put*, *to put on* and *to put across* are considered as three different verbs, and sentence *John puts on a hat* returns the triple $\langle$John,put_on,hat$\rangle$, in which *John* and *hat* are concepts associated to attribute *put_on*, indicating that *John* can *put_on* something, while a *hat* can be *put_on*). Triples/sentences involving verb 'to be' or nouns with adjectives provide immediate hints to build the sub-class structure in the taxonomy: for instance, "The dog is a domestic animal..." yields the relationships is_a(dog, animal) and is_a(domestic_animal,animal). In this way we have two kind of edges among nodes in the graph: verbal ones, labelled with the verb linking the two concepts and encoding the assertional knowledge, and taxonomic (is_a) ones encoding the definitional knowledge. Moreover, with the aim to enrich the representation formalism previously defined, we analysed the syntactic tree to seize the sentence positive or negative form based on the absence or presence (respectively) of a *negation modifier* for the verb.

In previous works, we focused on taxonomy construction (paying attention only to the definitional portion of the network), even though we already were building a semantic network. In this work we put the emphasis on the whole network (considering also its assertional portion), even though in the non-taxonomic portion of the network, also contingent knowledge is encoded.

3.2 Relational Pairwise Clustering

While [8] adopted a pairwise clustering technique based on the *Hamming distance* on the feature vector of each concept, here we propose to use a relational representation for concepts, and adopt the relational similarity function presented in [4]. In order to perform the proposed clustering method, the conceptual graph has been translated in first order logic, using the relations as binary predicates and the involved concepts as arguments. More precisely, we encoded the direction of the relation in the order of the arguments (the subject in the first place

Algorithm 1 Relational pairwise clustering of all concepts in the network.

Input: O is the set of objects (concepts) represented as in Section 3.2; T is the threshold for similarity function.
Output: set of clusters.

$pairs \leftarrow empty$
$averages \leftarrow empty$
for all $O_i \mid i \in O$ **do**
  $newCluster \leftarrow O_i$
  $clusters.add(newCluster)$
end for
for all $pair(C_k, C_z) \mid C \in clusters \wedge k, z \in [0, clusters.size[$ **do**
  if $completeLink(C_k, C_z, T)$ **then**
    $pairs.add(C_k, C_z)$
    $averages.add(getScoreAverage(C_k, C_z))$
  end if
end for
$pair \leftarrow getBestPair(pairs, averages)$
$merge(pair)$

completeLink(arg1,arg2,arg3) $\rightarrow$ TRUE if complete link assumption for the passed clusters holds, FALSE otherwise.
getBestPair(arg1,arg2) $\rightarrow$ returns the pair having the maximum average.

and the complement in the second), and the (positive or negative) valence of the action in the predicate name. Before translating the conceptual graph in a set of concepts described by the neighbors until a chosen radius, we extracted the weak components of the graph by JUNG in order to process the concepts in each separate sub-graph. A weak component is defined as a maximal sub-graph in which all its pairs of vertices are reachable from one another in the underlying undirected sub-graph. For each concept we extract the k-neighborhood around it, defined as the sub-graph induced by the set of concepts that are k or fewer hops away from the node. A *vertex/concept-induced sub-graph* is a subset of the vertices/concepts of a graph together with any edges whose endpoints are both in this subset. In this way we have a sub-graph of concepts for which exist at least one path long at most k hops between them and the root concept together with all links between each pair of concepts in this graph. This choice enriches the description language because we can portray a concept not only with the concepts along the considered paths, but with the whole graph of relations between its neighbours.

In practice, the sub-graph obtained for each concept X was translated into a Horn clause of the form $concepts(X) : -rel_a(X, Y), rel_b(Z, X), rel_c(Y, T)$, where Y, Z, T are the neighborhood of X and the binary predicates *rel* are verbs (positive or negative).

Pairwise clustering under the *complete link* assumption is applied to these descriptions: initially, each concept becomes a singleton cluster; then, clusters are merged while a merging condition is fulfilled (Algorithm 1). Complete link states that *the distance of the farthest items of the involved clusters must be less than a given threshold.*

3.3 Generalization of Cluster

In [8] the generalization task has been tackled taking advantage from an external resource. Unfortunately, for specific domains it is often unavailable. Then we need an alternative to overcome this limitation. Our proposal is to generalize each cluster using the maximum set of common descriptors of each concept. The generated concept will be the *subsumer* of the cluster. If the generated subsumer matches with an existing item, then it is promoted as subsumer, otherwise the generated concept remains without a human understandable name until a new concept will not match with it.

In order to perform the generalization phase, we apply the generalization operator proposed in [16], obtaining *least general generalizations*. For the sake of clarity, we cite its original definition:
A least general generalization (lgg) under $\theta_{OI} - subsumption$ of two clauses is a generalization which is not more general than any other such generalization, that is, it is either more specific than or not comparable to any other such generalization. Formally, given two Datalog clauses C_1 and C_2, C is a lgg under $\theta_{OI} - subsumption$ of C_1 and C_2 iff:

1. $C_i \leq_{OI} C, i = 1, 2$
2. $\forall D \; s.t. \; C_i \leq_{OI} D, i = 1, 2 : not(D <_{OI} C)$

$$\begin{aligned} lgg_{OI}(C_1, C_2) = \{C \mid\ & C_i \leq_{OI} C, i = 1, 2 \; and \\ & \forall D \; s.t. \; C_i \leq_{OI} D, i = 1, 2 : not(D <_{OI} C)\} \end{aligned}$$

The application of this operator opens to several conceptual graph refinements. In this work we focus on the insertion of new taxonomical relations. In some cases this leads to the bridging of potentially disjoint portion of the graph, but are exploitable for tasks as retrieval of documents of interest, as well as for the shifting of the representation when needed (abstraction).

3.4 Probabilistic Reasoning by Association

Several reasoning strategies can be exploited to extract novel information from the formalized knowledge. In particular, this framework allow to reason with the extracted knowledge, in the sense of finding a path of pairwise related concepts that establishes an indirect interaction between two concepts c' and c''. Since real world data are typically noisy and uncertain, there is the need for strategies that soften the classical rigid logical reasoning. Hence we keep soft relationships among concepts rather than hard ones, by weighting the relationships among concepts, where each arc/relationship is associated to a weight that represents its likelihood among all *possible worlds*. Thus we deploy two reasoning strategies: the former works in breadth and aims at obtaining the minimal path between concepts together with all involved relations, the latter works in depth and exploits Problog [15] in order to allow probabilistic queries on the conceptual graph. In more details the former strategy looks for a minimal path using a *Breadth-First Search* (BFS) technique, applied to both concepts under consideration. It also provides the number of positive/negative instances, and the corresponding ratios over the total, in order to help understanding different gradations (such as permitted, prohibited, typical, rare, etc.) of actions between two objects. While this value does not affect the reasoning strategy, it allows to distinguish which reasoning path is more suitable for a given task. Conversely, the latter strategy exploits the previous values in order to prefer some paths to other ones. It exploits ProbLog for this purpose, whose descriptions are based on the formalism $p_i :: f_i$ where f_i is a ground literal having probability p_i. In our case, f_i is of the form *link*(*subject*, *verb*, *complement*) and p_i is the ratio between the sum of all examples for which f_i holds and the sum of all possible links between *subject* and *complement*.

4 Evaluation

The proposed approach has been evaluated with the aim to obtain qualitative outcomes that may indicate its strengths and weaknesses. We exploited a dataset made up of documents concerning *social networks* on socio-political and economic topic, including 695 concepts and 727 relations. The size of the dataset was deliberately kept small in order to have poor knowledge. The similarity function ranges in $]0, 4[$. We executed several experiments varying the used threshold between $[2.0, 2.3]$ with hops equal to 0.5.

The experiments concern the qualitative examination of the obtained clusters and their generalizations, with the aim to understand the behaviour of this approach using a limited collection of documents. For lack of space, we show in Table 1 only the results for the threshold 2.0. We have chosen it because summarizes the largest set of clusters. The outcomes arising from the other thresholds do not show a qualitative difference, but just a quantitative one. Analysing the Table 1, we can see that several clusters seems to be unreliable. For this reason we have inspected some outcomes in order to understand what conditions led to this results. Each case that we present is special in some way.

Let to examine the outcome 35. Applying the lgg_{OI} we obtain:

$$\begin{aligned} concept(X) :- \ & impact(Y, X), signal(Y, X), signal\_as(Y, X), do\_with(Y, X), \\ & consider(Y, X), offer(Y, X), offer\_to(Y, X), average(Y, X), \\ & average\_about(Y, X), experience(Y, X), flee\_in(Y, X), be(Y, X). \end{aligned}$$

$$\theta = < \{internet/Y, visible/X\}, \{internet/Y, textual/X\} >$$

Through the substitution θ we find exactly the cluster $\{visible, textual\}$, that although seems to be unreliable, shows that in social network domain has been created on the bases of the full identity of its relations, with the same concept *internet*. This can be considered a special case because neither of the items can be promoted to subsumer, then we must leave unlabelled the subsumer.

Table 1. Clusters obtained processing concepts described using one level of their neighbourhood with a similarity threshold equal to 2.0.

#	Cluster
1	peer, preference, picture, thing, able, close, adopter, music, life
2	content, internet
3	way, computer
4	matter, online
5	television, school, facebook
6	domain, supervision
7	screen, broadband
8	adult, transition, fourteen
9	tool, point
10	hour, time
11	paramount, today
12	percent, household
13	interest, challenge
14	role, student

#	Cluster
15	property, executive, trouble, york, europe, fortune
16	segment, hand, america
17	relationship, capital, technology
18	question, game
19	income, female, newspaper, age
20	kid, guru, limit
21	theme, staff
22	classmate, note, connection, pass
23	magnet, comparison, relic
24	evolve, elite
25	day, view, modicum, community, station, clip, opportunity

#	Cluster
26	environment, activity
27	average, bulk, space, creator
28	conversation, educator
29	power, payoff, classroom, distraction, unfair, media
30	value, human
31	creation, announce, july
32	myspace, destination
33	phone, visitor, letter, class, touch
34	trivial, vetere, sincere, genuine
35	visible, textual
36	shift, collection
37	desire, sphere
38	future, gain

Now, let to consider the outcome 38. We can isolate as subsumer the concept:

$$\begin{aligned} concept(X) :- \; & lead(Y,X), lead\_into(Y,X), come\_for(Y,X), come\_to(Y,X), \\ & likely\_for(Y,X), be(Y,X), transform\_into(Y,X). \end{aligned}$$

$$\theta = < \{youth/Y, future/X\}, \{youth/Y, gain/X\} >$$

Using the substitution θ we obtain the cluster $\{future, gain\}$, again describing their role with respect to the single concept *youth*. We want to point out that the portion of *future* left out from the lgg_{OI} does not regard *youth*. The shared description that determines their similarity regards their relations with the same concept *youth* and nothing else, reinforcing their correlation. Follows the uncovered portion of *future*.

$$\begin{aligned} & take\_into(plan, future), be(future, digital), see\_in(negotiation, future), \\ & take(plan, future), be(negotiation, future), see(negotiation, future). \end{aligned}$$

In this case we obtained a full mapping with the single item *gain*, that can label the subsumer as best representative of the cluster.

Finally, let to analyse the cluster 20.

$$\begin{aligned} concept(X) :- \; & protect(Y,X), protect\_by(Y,X), become(Y,X), use(Y,X), \\ & have(Y,X), have\_to(Y,X), have\_in(Y,X), have\_on(Y,X), \\ & find(Y,X), go(Y,X), look(Y,X), begin(Y,X), begin\_with(Y,X), \\ & begin\_about(Y,X), suspect\_in(Y,X), suspect\_for(Y,X). \end{aligned}$$

$$\theta = < \{\, parent/Y, kid/X\}, \{parent/Y, guru/X\}, \{parent/Y, limit/X\} >$$

The substitution θ do not fully cover any item of the original cluster. Indeed, for each concept a partial description has been left out. As for the first case, the generated concept needs to stay unlabelled. In particular, to obtain *kid* we need to add:

$teach(kid, school), launch\_about(foundation, kid), teach(kid, contrast),$
$come\_from(kid, contrast), launch(foundation, kid), finish\_in(kid, school),$
$invite\_from(school, parent), possess\_to(school, parent), invite(school, parent),$
$finish\_in(kid, side), come\_from(kid, school), find\_in(school, parent),$
$produce(school, parent), come\_from(kid, side), find\_from(school, parent),$
$finish\_in(kid, contrast), invite\_about(school, parent), come\_before(school, parent),$
$release(foundation, kid), invite\_to(school, parent), teach(kid, side),$
$release\_from(foundation, kid).$

The same type of addition is needed for *guru*:

$$become(teenager, guru).$$

Lastly, we have *limit* through the addition of:

$$be(ability, limit), limit(ability, limit).$$

The results show that the procedure seems to be reliable in order to recognize similar concepts on the basis of their structural position in the graph. Then we believe that applying this approach to more than one level of description, can be achieved interesting results. The improvement performed in this work can be appreciated remarking the novelty in the method of description construction. Using the Hamming distance we obtained a first level *relation centric* (i.e. that describes the concept with its direct relations), meanwhile using our method we obtain a *concept centric* description (i.e. using direct and indirect relations between the first level concepts).

5 Conclusions

This work proposes a technique to automatically lean conceptual graphs from text, avoiding the support of external resources. We apply a relational clustering to group similar concepts, and then generalize the items in each cluster to obtain new concepts that can be used to build taxonomic relations in the graph, in some cases bridging disjoint portions of it. As planned in previous works, we used a technique of anaphora resolution to improve the quality of the semantic network. Examining the results, we have seen that the restricted collection regarding a specific domain on which the conceptual graph has been built, has affected the results. In any case, the procedure seems to be reliable in order to recognize similar concepts on the basis of their structural position in the graph. As further studies, we plan to improve the method of building of the graph, and to run further experiments representing the concepts with more than one level of neighbours.

Acknowledgments

This work was partially funded by Italian FAR project DM19410 MBLab "Laboratorio di Bioinformatica per la Biodiversit Molecolare" and Italian PON 2007-2013 project PON02_00563_3489339 "Puglia@Service".

References

[1] P. Cimiano, A. Hotho, and S. Staab. Learning concept hierarchies from text corpora using formal concept analysis. *J. Artif. Int. Res.*, 24(1):305–339.

[2] M.C. de Marneffe, B. MacCartney, and C. D. Manning. Generating typed dependency parses from phrase structure trees. In *LREC*, 2006.

[3] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database.* MIT Press, Cambridge, MA, 1998.

[4] S. Ferilli, T. M. A. Basile, M. Biba, N. Di Mauro, and F. Esposito. A general similarity framework for horn clause logic. *Fundam. Inf.*, 90(1-2):43–66, January 2009.

[5] Svetlana Hensman. Construction of conceptual graph representation of texts. In *Proc. of the Student Research Workshop at HLT-NAACL 2004*, HLT-SRWS '04, pages 49–54, Stroudsburg, PA, USA, 2004. ACL.

[6] K. Kipper, H. T. Dang, and M. Palmer. Class-based construction of a verb lexicon. In *Proc. of the 17th NCAI and 12th IAAI Conference*, pages 691–696. AAAI Press, 2000.

[7] D. Klein and C. D. Manning. Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2003.

[8] F. Leuzzi, S. Ferilli, C. Taranto, and F. Rotella. Improving robustness and flexibility of concept taxonomy learning from text. In *In Proc. of The Workshop on NFMCP at ECML-PKDD 2012 Conference*, 2012.

[9] A. Maedche and S. Staab. Mining ontologies from text. In *EKAW*, pages 189–202, 2000.

[10] A. Maedche and S. Staab. The text-to-onto ontology learning environment. In *ICCS-2000 - Eight ICCS, Software Demonstration*, 2000.

[11] N. Ogata. A formal ontology discovery from web documents. In *Web Intelligence: R.D., First Asia-Pacific Conf. (WI 2001)*, number 2198 in LNAI, pages 514–519. Springer-Verlag, 2001.

[12] J. O'Madadhain, D. Fisher, S. White, and Y. Boey. The JUNG (Java Universal Network/Graph) Framework. Technical report, UCI-ICS, October 2003.

[13] A. Cucchiarelli P. Velardi, R. Navigli and F. Neri. Evaluation of OntoLearn, a methodology for automatic population of domain ontologies. In *Ontology Learning from Text: Methods, Applications and Evaluation.* IOS Press, 2006.

[14] L Qiu, M.Y. Kan, and T.S. Chua. A public reference implementation of the rap anaphora resolution algorithm. In *Proceedings of LREC 2004*, pages 291–294, 2004.

[15] L. De Raedt, A. Kimmig, and H. Toivonen. Problog: a probabilistic prolog and its application in link discovery. In *In Proc. of 20th IJCAI*, pages 2468–2473. AAAI Press, 2007.

[16] G. Semeraro, F. Esposito, D. Malerba, N. Fanizzi, and S. Ferilli. A logic framework for the incremental inductive synthesis of datalog theories. In Norbert E. Fuchs, editor, *LOPSTR*, volume 1463 of *LNCS*, pages 300–321. Springer, 1997.