Discovering Semantic Relations within Nominals

Mihaela Colhon[1], Dan Cristea[2,3], Daniela Gîfu[2]

[1]Department of Computer Science, University of Craiova
[2]Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași
[3]Institute for Computer Science Romanian Academy - the Iași branch

mcolhon@inf.ucv.ro,{dcristea,daniela.gifu}@info.uaic.ro

Abstract. We are interested to develop a technology able to discover entities and relations connecting them, as expressed in fiction texts. Deciphering these links is a major step in understanding the content of books. In this study we consider the case of imbricated entities, therefore entities realized at the surface text level by imbricated spans. For this research we use the *QuoVadis* corpus, whose conventions of annotations we describe briefly, same as some statistics on the types of relations, features regarding the relations' arguments and words or expressions functioning as triggers. The approach to recognize the semantic relations is based on patterns extracted from the corpus. The evaluation shows very promising results.

Keywords: semantic relations, nominal expressions, annotated corpus, anaphora, annotation conventions

1 Introduction

Extracting relations among entities is an active research area in the field of linked textual data, with applications in the areas of semantic and social web [6], information extraction [26] and text mining. In the last years, the search engines usage for recognizing the relation between two entities has also gained attention [13]. Text mining implies meaningful representation of texts, including encoding of entities and relations occurring between them.

The field touches the very essence of the deep understanding of natural language. As language exhibits a huge diversity of expression for entities as well as relations connecting them, any consideration here should be made on a firm representational ground. Thus, the main major preoccupation for NLP technologies related to text mining (the enlarged field of the old information extraction area) goes in the following directions: 1. finding sound representations at a conceptual level; 2. decoding language onto this representation, and 3. mimicking the reasoning capacity of humans, which is manifested in our skills to understand and make use of the language in real life.

In this study we are mainly concerned with parts of the second topic, namely to find ways to map the huge diversity of natural language expressions onto sound conceptual level representations. Our annotations are meant to show simultaneously the constructions at the basic language level and the equivalent encodings at the knowledge representation level (entities and relations between them). If, in general, entities are of a very diverse nature: persons, animals, places, organizations, crafts, objects, ideas, events, moments of time, etc., in this study we are concerned only with persons, gods and any groupings of them. Among the extremely large class of semantic relations that a text could express, we decipher four types: anaphoric (or referential) relations (when the interpretation of one entity mention is dependent on the interpretation of a previous one), affectional relations (when a certain feeling or emotion, is expressed in the interaction of characters), kinship relations (when family relationships are mentioned, sometimes composing very complex genealogical trees), social relations (when job hierarchies or social mutual ranks are explicitly remarked) [5].

Syntactically, the text realization of entities is nominal phrases (NPs). However, NPs have sometimes recursive structures, such that one NP may include one or more other NPs. Since to each NP at the textual level, an entity is mapped at the representational level, the recursiveness of the NP chunks is reflected by relations between corresponding entities. Some examples are: University of Washington (mentioning an institution, a location and the positioning of the institution in that location) or mother of the child (including two entities of type person: the mother of the child and the child itself and a kinship relation linking them). To reflect the surface structure, these entities are often referred to as nested or imbricated [8]. It should be noted that imbricated NPs have always separate heads and there are not NPs that intersect and are non-imbricated [5].

In this paper we propose a method for automatic recognition of semantic relations that are mentioned within imbricated entities. The method uses lexico-syntactic patterns extracted from a training corpus, which are constructed by exploring the complex annotations of the lexical resource.

The paper is organized as follows. In the following sections we give an overview of the similar studies in name entity recognition, relations discovery and nested semantic relations. Section 4 describes our approach for automatic relation recognition within nominals. Section 5 presents evaluation considerations and the last section makes concluding remarks and presents ways of improving this recognition mechanism.

2 Related work

Most of the designers of corpora dedicated to named entity recognition (NER) usually ignore nested entities by choosing to focus on the outermost entities and proposing in this manner flat entity representations. The widely used MUC-6, and MUC-7 NER corpora, composed of American and British newswire, are flatly annotated. The GENIA corpus [3] contains biomedical named entities, but the JNLPBA 2004 shared task [4], which utilized the corpus, removed all embedded entities for the evaluation [8]. To our knowledge, the only shared task which has included nested entities for evaluation is the SemEval 2007 Task 9 [18], which used a subset of the AnCora corpus.

NLP researchers use corpora annotated with semantic links for training recognition algorithms. Since the 80s, the normalized for nested relations and object-oriented database become objectives for many researchers [21]. A considerable number of studies are concentrated on nested relational normal forms, like: NNF [21], NF2 [22] and NF-NR [16].

Özsoyoglu and Yuan, in NNF (Nested Normal Form), consider that the nested relations are structured as trees, called scheme trees, and introduce a normal form for these relations, called the nested normal form. The representation of nested sets as trees or hierarchies we find it in Nested set model, sometimes with different names like Recursive Hierarchies [12].

NF2 model is, actually, an extension of the classical relational model, which focuses on relation-valued attributes, improving with a reformulation of query operations of the frame model in terms of NF2 algebra operations [22].

NF–NR model removes inconvenient anomalies from a nested relational database schema, such as global redundancies between nested relations [16], considering two approaches: the restructuring the nested relations by applying a set of rules that transform relations NF-NR nested relations and the entity-relationship to NF-NR database design, based on the normal form for ER model [15]. Moreover, the nested relations discusses as a database model, are also research topics for many researchers [11, 17, 23, 24], emphasizing the ability to represent and manipulate complex structures [1].

Furthermore, in literature, a few researchers have focused on introducing imprecise and uncertain information into NF relational database, in so-called the fuzzy nested relational models. We remind a NF database model with null values [14] or the modeling in NF data model of the uncertain null values, set values, range values, and uncertain values, being extended NF algebra on similarity-based [3].

In addition, to define and recognize nested relations, he worked with Relix (RElational database programming Language in UNIX), a system focused on two kinds of data models: domains and relations [10] and the nested relations were built on top of relations and nested queries by allowing the domain algebra to subsume the relational algebra. In this paper, we have focused on syntactic pattern, an approach that has been used. For instance, QBQL syntax, which has built-in set join operations, nested relations provide an alternative.

3 Our previous work

Our work continues the research described in [5], and summarized in this section. The essence of the research aimed at building a corpus of annotated entities and semantic relations. The text used was the Romanian version of the novel "Quo Vadis", authored by the Nobel laureate Henryk Sienkiewicz[1]. We have presented in the mentioned paper the marking conventions, designed to incorporate annotations for persons

[1] Version translated by Remus Luca and Elena Linţă and published at Tenzi Publishing House in 1991. The aligned passages are searched in the English translation made by Jeremiah Curtin and published by Little Brown and Company in 1897.

and god type entities, including groups, and for relations linking them. The annotation itself was a time consuming and painful process, that run over more than two years.

3.1 Annotating semantic relations

Out of the vast variety of relations that could come together with the mentions of characters in fiction texts, we have concentrated on 12 types of anaphoric relations (examples are: `coref`, `member-of`, `part-of`, `has-as-part`, etc.) and almost 30 types of non-anaphoric relations (examples are: `parent-of`, `child-of`, `love`, `friendship`, `hate`, `superior-of`, `inferior-of`, `colleague-of`, etc.). We believe this set of relations covers to a large extend the relational inventory in a fiction text. Each segment of text received by annotators included already basic markings on token/part-of-speech/lemma layers, performed automatically during a pre-processing phase [25]. Based on these, the manual annotation captured the following aspects: notation of mention of entity, the relation's boundaries, the type of relation, its two arguments (all relations are binary) and, where present, the trigger (a word or an expression signaling the relation). All the files contributed by individual annotators were then merged automatically in a contiguous file, IDs of XML elements and their references being re-generated. To this initial level of annotation we added, recently, syntactic dependency data: each token of all sentences has been complemented with its head-word and the dependency relation towards the head.

The corpus thus obtained was used to train a process able to generalize patterns and to identify features for automatic relation discovery. At this stage of the research, we have concentrated only on the automatic recognition of the non-anaphoric relations, also ignoring the identification of entities.

At the text level, the entities are realized[2] as noun phrases whose heads are nouns or pronouns. These constructions may include, besides their heads, also modifiers, such as determiners, adjectives, numerals, genitival constructions, or prepositional phrases. However, we imposed the constraint that noun phrases do not extend over relative clauses.

Considering the relative positioning in the text of the spans of text that realize the entities forming the two arguments of the relations, they could intersect or not. If they intersect, then (empirical evidence show that) they are necessarily nested (imbricated) and the convention for the direction of the relation is to consider as FROM the larger entity and as TO the nested entity [5]. For instance (here and below, entities are marked in square brackets and triggers in angular brackets):

`1:[<copilul> drag 2:[al celebrului Aulus]]` (in the English version, `1:[a dear <child> of 2:[the famous Aulus]]`)

In this example, the FROM entity noted here with [1] imbricates the TO entity denoted by [2] and there is a child-of relation between [1] and [2]. The trigger here is `<copilul>` (EN: `<child>`).

In this study, we explore only the relations occurring between nested entities.

[2] The term “realize” is the one used in Centering [9].

3.2 The inventory of semantic relations

As noted in the literature [7], there is not an universally accepted list of semantic relations to be considered between nominal groups. Different teams of researchers consider their particular lists, very much dependent on the domain of the analyzed texts of the application envisioned. In [5] the semantic relations marked in the *QuoVadis* corpus are grouped in four classes, and the same classification applies to imbricated entities. Some examples follow:

– relations of the class AFFECT:

1:[a <prietenului>, amicului şi confidentului 2:[lui Nero]] => AFFECT. friend-of (EN: 1:[of 2:[Nero]'s <friend>, companion, and suggester]).
1:[<favoritul> 2:[împăratului]] => AFFECT.loved-by (EN: 1:[2:[Cæsar]'s first <favorite>]).
1:[fosta <amantă> 2:[a lui Nero]] => AFFECT.rec-love (EN: 1:[the former <favorite>2:[of Nero]]).
1:[unui <credincios> 2:[al "Mielului"]] => AFFECT.worship (EN: 1:[a <confessor> 2:[of the "Lamb"]]).

– relations of the class KINSHIP:

1:[propriul 2:[lor] <copil>] => KINSHIP.child-of (EN: 1:[their 2:[own] <daughter>]).
1:[scumpii 2:[săi] <nepoţi>] => KINSHIP.nephew-of (EN: 1:[2:[his] dear <nephews>]).
1:[<părinte> 2:[al zeilor]] => KINSHIP.parent-of (EN: 1:[<father 2:[of the gods]]).
1:[nişte <fraţi> 2:[ai tăi]] => KINSHIP.sibling (EN: 1:[thy 2:[own] <brothers>]]).
1:[nefericita <soţie> 2:[a lui Zethos]] => KINSHIP.spouse-of (EN: 1:[the unhappy <wife> 2:[of Zethos]]).
1:[furtunos <urmaş> 2:[al consulilor]] => KINSHIP.unknown (EN: 1:[mad <descendant> 2:[of consuls]]).

– relations of the class SOCIAL:

1:[<tovarăşul> 2:[lui Petru]] => SOCIAL.colleague-of (EN: 1:[2:[Peter's] <companion>]]).
1:[<adversarului> 2:[său] greoi] => SOCIAL.in-competition-with (EN: 1:[2:[his] heavy <antagonist>]]).
1:[a tuturor <sclavilor> 2:[prefectului Pedanius Secundus]] => SOCIAL.inferior-of (EN: 1:[all the <slaves> 2:[of the prefect Pedanius Secundus]]).
1:[cei mai aprigi <duşmani> 2:[ai Romei]] => SOCIAL.opposite-to (EN: 1:[2:[of Rome's] most inveterate <enemies>]]).

1:[<comandanţi> de 2:[cohorte]] => SOCIAL.superior-of (EN: 1:[2:[pretorian] <leaders>).

– relations of the class REFERENTIAL:

1:[<unul dintre> 2:[sclavi]] => REFERENTIAL.member-of (EN: 1:[one of my 2:[slaves]]).
1:[Apostolul cu 2:[<barba> argintie]] => REFERENTIAL.has-as-part (EN: 1:[Apostle with his 2:[silvery beard]])[3].
1:[<numele> de 2:[roman]] REFERENTIAL.name-of (EN: 1:[2:[Roman] name]).
1:[<faţa> 2:[ei] tristă, dar senină] REFERENTIAL.part-of (EN: 1:[2:[her] face, pensive, but mild]).
1:[un <grup> dintre 2:[celelalte slugi]] REFERENTIAL.subgroup-of (EN: 1:[a crowd of 2:[other servants]]).

4 Automatic recognition of semantic relations

Syntactically, the text realization of entities is nominal phrases (NPs). As noted in literature, entity identification and relation extraction are two separate tasks, the second one actually following the first one [20]. We relied, in our study, on the manual annotations for entities in the *QuoVadis* corpus, and focused our attention strictly on the automatic identification of the semantic relations between nested entities. In the following section, we describe two approaches, one using morpho-syntactic information, the other - dependency data, i.e., conforming to [19], asymmetrical functional relations between pairs of words, considered head and modifier.

4.1 Morpho-syntactic patterns

A collection of morpho-syntactic patterns has been extracted from the occurrences of imbricated relation spans belonging to the training corpus. Sequences of items covering the whole span of the relation sharing similar morpho-syntactic structures can be considered candidates for the corresponding type of relation. No lexical information is considered here yet, although we are aware that it has an important role in identification of relations. The main scope of this approach is to generalize the syntactic patterns found in the corpus under the same relation realization in order to discover similar sequences, instantiated or not in the corpus, which could belong to the same relation type. To allow a larger flexibility, generalizations of patterns can also be easily mastered by including wildcards.

Let us consider the following example:

1:[2:[împăratul însuşi], şi 3:[preoţii], şi 4:[vestalele], şi 5:[senatorii], şi 6:[cavalerii], şi

[3] The annotations for the English sequences copy those in the Romanian versions.

`7:[poporul]]` (EN: `1:[2:[Cæsar himself] bet; 3:[priests], 4:[vestals], 5:[senators], 6:[knights] bet; 7:[the populace] bet)`.

Here, the entity on the position of FROM argument, [1], is in a `REFERENTIAL.has-as-member` relation with the entity [2] and with a `REFERENTIAL.has-as-subgroup` relation with each of the entities [3], [4], [5], [6] and [7]. The morpho-syntactic structure of the text corresponding to the entity [1] shows an enumeration of noun phrases separated in the text by the connectors Cc (for conjunction) and COMMA (for comma):

`1:[2:[Ncmsry Dh3ms] COMMA Cc 3:[Ncmpry] COMMA Cc 4:[Ncfpry] COMMA Cc 5:[Ncmpry] COMMA Cc 6:[Ncmpry] COMMA Cc 7:[Ncmsry]].`

At this stage of our study, morpho-syntactic patterns are only used to detect `REFERENTIAL` relations realized by enumeration sequences, from an external PERSON-GROUP entity to imbricated PERSON and PERSON-GROUP entities. However, as can be noticed from this example, without lexical information that identifies the semantic class of inner entities no distinction can be made among the class of relations that share the same general structure of the two arguments, namely FROM being a group entity and TO being different types of components: member, part or sub-part. These relations are, correspondingly: `has-as-member`, `has-as-part` and `has-as-subgroup`.

4.2 Lexical-dependency patterns

In this approach, the text of the sentence that includes the span of the relation is extracted and processed with a Dependency Parser and the resulted dependency links relating the arguments' spans are put in evidence. Here, patterns are generated as a sequence of typed dependencies linking the lemma of the head of the inner entity to the lemma of the head of the outermost entities. For instance, in the following example, where heads are underlined, a `KINSHIP.unknown` relation is established between the entities [1] and [2], triggered by `<urmaș>` (EN: `<descendant>`), which is the head word of entity [1].

`1:[furtunos <urmaș> 2:[al consulilor]]` (EN: `1:[mad <descendant> 2:[of consuls]])`

At the dependency level there is a substantive attribute (a.subst) relation linking the head of [2], *consulilor* (*consuls*), with the head of the entity [1], *urmaș* (*descendant*), as is illustrated in Fig.1.

An empirical analysis of the cases encountered in the training corpus resulted in the following steady observations, put here as hypotheses:

Hypothesis 1: The outer-inner imbrication of the text spans corresponding to the two arguments of an imbricated semantic relation is copied also on the dependency tree, where the tree structure corresponding to the inner span is a sub-tree of the one corresponding to the outer span.

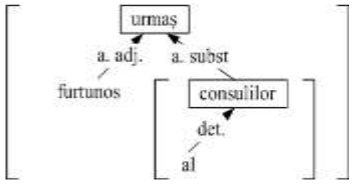

Fig. 1. The dependency parse tree for the text *furtunos urmaș al consulilor* (EN: *mad descendant of consuls*) – example for the a.subst relation

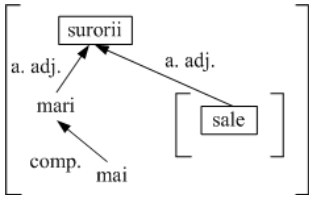

Fig. 2 The dependency parse tree for the text *surorii sale mai mari* (EN: *to her elder sister*) – example for the a.adj relation

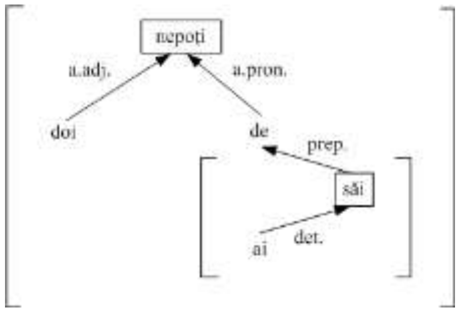

Fig. 3. The dependency parse tree for the text *doi nepoți de ai săi* (approx. EN: *two of his nephews*) – example for the a.adj relation

Hypothesis 2: If the direction of a relation in the syntactic tree is considered from a word towards its dependency head (parent), then there is always a path of dependency relations linking the NP head of the inner entity to the NP head of the outer entity.[4]

[4] Care should be taken when using the term 'head': in the context of the surface text, we say that a group (for instance, an NP) has a head; in the context of a dependency tree, we say that each word of a sentence (excepting the main verb), has a head. When confusion could occur, we will use the expressions 'NP head' for the first case, and 'dependency head' – for

All entities in our corpus are realized by nominal phrases (head being the central noun of the group) or by pronominal phrases (head being the central pronoun). Three types of dependency relations link the head of the inner entity to the head of the outer entity.

- substantival adjunct (a.subst) – when the head of the inner entity is a noun. The example in Figure 1 shows the dependency tree for 1:[furtunos <urmaș> 2:[al consulilor]], where a KINSHIP.unknown relation holds between the outer and the inner entities.

- adjectival adjunct (a.adj) – when the head of the inner entity is a numeral or an adjectival pronoun (in this case there is an agreement in gender, number and case between the two heads). Figure 2 shows the construction 1:[<surorii> 2:[sale] mai mari], which has the morpho-syntactic pattern 1:[<Ncfsoy> 2:[Ds3fsos] Rg Afpfprn].

- pronominal adjunct (a.pron) – when the head of the inner entity is a pronoun (and there is no agreement between the two heads). Figure 3 shows a KINSHIP.nephew-of relation within the construction 1:[doi <nepoţi> de-2:[ai săi]], which has the morpho-syntactic pattern 1:[Mcmp-l Ncmprn Sp 2:[Tsmpr Ps3mp-s]].

Hypothesis 3: The trigger of the semantic relation includes (if it is not identical with) the NP head word of the outer entity.

Now, considering the case of two imbricated NPs at the text level, a corollary could be drawn out of the first two hypotheses:

Corollary: A semantic relation between two imbricated entities corresponds to the inverse of a path of dependency relations linking the NP heads of the two entities.

Our recognition algorithm exploits this observation and the lexicalization of the trigger. We used this corollary to prepare an external resource that helped in the recognition process, a process that can be described as here:

- a list of triggers, under the form of lemmas, was created for each type+subtype of a semantic relation identified in the corpus (Appendix A lists the identified triggers in the *QuoVadis* corpus, corresponding to the semantic relations annotated between nested entities);

- a dependency parser was run on the relation spans (the outer entities) on the entire corpus. As a result, the dependency trees of these sub-sentences were generated, the head words of the larger and inner entities were localized, and the dependency path linking the two heads were extracted;

---

the second. As such, an NP head is a word belonging to an NP and, seen as participant in a dependency tree, itself has a dependency head, which is external to the NP.

– out of the paths extracted at step 2, those corresponding to annotated outer-inner entities linked by semantic relations were selected and this list was sorted by the semantic relation's type+subtype.

The resources resulted at steps 1 and 3 were merged and grouped by type+subtype of the semantic relations, resulting in a mapping function: $T \rightarrow 2L \times P$, where T is a combination type+subtype, L is a list of trigger lemmas and P is a list of paths of dependency relations. Moreover, the elements of L and P were sorted from the most frequent to the less frequent for each type+subtype of T.

With this resource, the recognition algorithm is an inverse function from $L \times P$ to T, trying to identify that combination type+subtype of a semantic relation which corresponds to the highest product of relative frequencies of the pair (l, p), where l is the lemma of the outer entity and p is a path including or equal to the one linking the inner entity to the outer entity on the dependency sub-tree. This algorithm indeed recognizes the relation type based on the trigger and the dependency structure.

5 Evaluation

The *QuoVadis* corpus contains 22,303 annotated relations, out of which 1,240 occur between nested entities. These pairs of entities were used for building the auxiliary resources and the patterns.

From the total number of relations annotated in the corpus we have kept 90% for training and 10% for testing, using a cross-validations policy, which guaranteed no intersection between training and evaluation sentences. It resulted a number of 1,116 relations used in the training phase (24 AFFECT relations, 182 SOCIAL relations, 153 KINSHIP relations and 757 REFERENTIAL relations) and 124 relations for testing (3 AFFECT relations, 20 SOCIAL relations, 16 KINSHIP relations and 85 REFERENTIAL relations).

The evaluation task considered here is this: the system is presented with an entity which nests at least one other entity (extracted from the manual annotations of the corpus) and has to decide whether a relation exists between the marked entities and, if yes, to identify its type and subtype.

Table 1. Evaluation scores for the recognition of semantic relations between nested entities

Relation type	**# test relations**	**# correct**	**Precision**	**Recall**	**F-measure**
AFFECT	27	23	0.9	0.86	0.87
KINSHIP	169	147	0.94	0.87	0.9
SOCIAL	202	155	0.92	0.76	0.83
REFEENTIAL	842	757	0.96	0.9	0.93
Total	1240	1082	0.93	0.85	0.88

As it can be seen, maximum precision and recall is obtained for those types which are sparsely represented in the corpus. We suspect that more training data is needed to get relevant figures. Moreover, the lowest F-measure stays with the `SOCIAL` class, which has the peculiarity of a very rich set of triggers for each subtype. This high diversification makes also less probable the occurrence of the same relation trigger in both the training and the test parts of the corpus.

The most common problems in recognition are caused by the fact that the head of the FROM entity is not found in any set of triggers extracted from the training corpus. Examples of misses are:

- `REFERENTIAL.part-of` relation in `1:[căpșorul 2:[fetei]]` (EN: `1:[2:[the maiden's] head]`)
- `REFERENTIAL.part-of` relation in `1:[carne de 2:[copii]]` (EN: `1:[flesh of 2:[children]]`)
- `SOCIAL.inferior-of` relation in `1:[unui arendaș 2:[al său]]` (EN: `1:[a confidant 2:[of Vinicius]]`)
- `SOCIAL.superior-of` relation in `1:[păzitor din oficiu 2:[al ostaticei]]` (EN: `1:[official guardian 2:[of the hostage]]`)
- `SOCIAL.superior-of` relation in `1:[acel pontifex maximus 2:[al creștinilor]]` (EN: `1:[that pontifex maximus 2:[of the Christians]]`)

There are also sequences where the head word of the FROM entity does not help in recognizing the relation, like in this case for `REFERENTIAL.has-name` relation, where two other relations are considered to occur: `REFERENTIAL.subgroup-of` and `SOCIAL.inferior-of`:

`1:[doi sclavi din neamul 2:[quazilor]]` (EN: `1:[two powerful 2:[Quadi]]`)

6 Discussions and Conclusions

This paper presents work in progress in the area of recognition of semantic relations that occur within nested textual entities, linguistically expressed as noun phrases. Although our immediate goal was to develop a tool that would help a user to interact with the *MappingBooks* application, the aim is more generous and the results overpass this limited horizon. We have exposed here a corpus developed in a previous project, conventions of annotation at the entities and semantic relations layers, heuristics for the extraction of semantic relations that make use of lexical and syntactic data, and previous results showing already a promising accuracy.

We are aware already of a number of problems that our approach could hide. One of them is, for example, how could enumerations (of elements in a set, or of parts of a whole, or of subsets belonging to a larger set) be distinguished from apposition or enumeration of qualities describing the same entity. An example is the following:

1:[a <prietenului>... și confidentului 2:[lui Nero]]
(EN: 1:[a <friend> and confident 2:[of Nero]]

where both terms of the conjunction refer the same person and an AFFECT.friend-of relation should be recognized between [1] and [2].

Another potential problem is given by the need to go beyond the lexical lists suggested by triggers as cues to distinguish different types of semantic relations, in the need to enhance the recall. Indeed, we are aware that going out of our annotated corpus, to other documents or genres, the recall will deteriorate drastically. To make the tool more reliable, sources that contain semantically related words, as WordNet, are needed. For instance, if the training corpus contains *soție* (EN: *wife*), it would be good to have also: *soață*, *nevastă*, *muiere*, *consoartă*, *femeie*, *tovarășă de viață*, *jumătate*, *pereche* etc. (synonyms and metaphors of *wife*).

Finally, ambiguities, therefore deteriorating the precision, are induced by triggers having more senses, as here:

1:[<capul> 2:[omului]] (EN: 1:[2:[man's] <head>]), inducing a REFERENTIAL.part-of relation

and here:

1:[<capul> 2:[familiei]] (EN: 1:[<the head> of 2:[the family]])

inducing a SOCIAL.superior-of relation. To face this problem, a word sense disambiguation phase should precede detection of relations.

Acknowledgements. This work was partly supported by *MappingBooks*, a project financed by the Romanian Ministry of Education and Research (UEFISCDI) under the Partnerships Programme (PN II Parteneriate, competition PCCA 2013), project code: PN-II-PT-PCCA-2013-4-1878. Thanks are addressed to Radu Simionescu for performing the dependency parsing of the *QuoVadis* corpus and all members of the NLP-Group@UAIC-FII which contributed to the creation of this corpus.

References

1. Abiteboul, S., Scholl, M.: From simple to sophisticated languages for complex objects. In Special Issue on Nested Relations. IEEE Data Engineering. Aug (1988)
2. Bibiri, A.D.: An Annotated Corpus of Entities and Semantic Relations. Dissertation thesis presented at The Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași (2014)
3. Buckles, B.P., Petry, F.: A fuzzy representation of data for relational databases. Fuzzy Sets Syst. 7, pp. 213–226 (1982)

4. Collier, N., Kim, J., Tateisi, Y., Ohta, T., Tsuruoka, Y. (eds).: Proceedings of the International Joint Workshop on NLP in Biomedicine and its Applications (2004)
5. Cristea, D., Gîfu, D., Colhon, M., Diac, P., Bibiri, A.-D., Mărănduc, C., Scutelnicu, L.-A.: Quo Vadis: A Corpus of Entities and Relations. In: Lexicon, Springer International Publishing Switzerland (2015)
6. Ding, L., Finin, T., Joshi, A., Peng, Y., Cost, R.S., Sachs, J., Pang, R., Reddivari, P., Doshi, V.: Swoogle: A Semantic Web Search And Metadata Engine. In: In Proceedings of the 13th ACM Conference on Information and Knowledge Management (2004)
7. Downing, P.: On the creation and use of English noun compounds. In: Language, 53(4), pp. 810-842 (1997) DOI: 10.2307/412913. 14, 18, 20
8. Finkel, J.R., Manning, C.D.: Nested Named Entity Recognition. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP '09), 141-150 (2009)
9. Grosz, B., Joshi, A., Weinstein, S.: Centering: A Framework for Modelling the Local Coherence of Discourse. In: Computational Linguistics, 2(21) (1995)
10. Hongbo, H.E: Implementation of Nested Relations in a Database Programming Language, Master Thesis, School of Computer Science McGill University, Montreal (1997)
11. Jaeschke, G., Schek, H. J.: Remarks on the Algebra of Non first Normal Form Relations. In: ACM Symposium on Principles of Database Systems. Los Angeles (1982)
12. Kamfonas, M.: Recursive hierarchies: the relational taboo!, The relational Journal, (1992)
13. Keller, F., Lapata, M.: Using the Web to Obtain Frequencies for Unseen Bigrams. In: Computational Linguistics 29(3), pp. 459–484 (2003)
14. Kim, J.-D. Ohta, T., Tateisi, Y., Tsujii, J.: GENIA corpus semantically annotated corpus for bio-textmining, Bioinformatics, 19, pp. 180-182 (2003), DOI: 10.1093/bioinformatics/btg102
15. Ling, T. W.: A Normal Form for Entity-Relationship Diagrams. In: Proceedings of 4th International Conference on Entity-Relationship Approach (1985).
16. Ling, T.W., Yan, L. L.: NF-NR: A practical Normal Form for Nested Relations. In: Journal of Systems Integration, 4, pp. 309-340 (1994)
17. Makinouchi, A.: A Consideration on Normal Form of Not-Necessarily Normalized Relation in the Relational Data Model. In: Proceedings of 3rd International Conference on VLDB. Japan (1977)
18. Márquez, L., Villarejo, L., Marti, M.A., Taulè, M.: Semeval-2007 task 09: Multilevel semantic annotation of Catalan and Spanish. In: Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval- 2007) (2007)
19. Mel'čuk, I. Dependency Theory: Syntax and Practice, Albany, NY:SUNY Press. (1987)
20. Năstase, V., Nakov, P., Ó. Séaghdha, D., Szpakowicz, S.: Semantic Relations Between Nominals. In: Morgan & Claypool Publishers, California (USA) (2013)
21. Özsoyoglu, Z. M., Yaun, L.-Y.: A new normal form for nested relations. In: ACM Transactions on Database Systems Journal, 21(1), pp.111-136 (1987)
22. Reimer, U.: A Frame-Based Knowledge Representation Model and its Mapping to Nested Relations. In: Data and Knowledge Engineering Journal, 12 (4), pp. 321-352 (1989)
23. Roth, M. A., Korth, H. F.: The Design of ¬1NF Relational Databases into Nested Normal Form. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 143-159 (1987)
24. Roth, M. A., Korth, H. F., Siberschatz, A.: Extended Algebra and Calculus for Nested Relational Databases. In: ACM Transaction on Database Systems Journal, 13(4), pp. 389-417 (1988)

25. Simionescu, R. Graphical grammar studio as a constraint grammar solution for part of speech tagging. In: M. A. Moruz, D. Cristea, D. Tufiş, A. Iftene, H. N. Teodorescu (eds.) Proceedings of the 8th International Conference "Linguistic Resources And Tools For Processing Of The Romanian Language", pp. 109-118 (2012)
26. Sheth, A.P., Ramakrishnan, C.: Relationship web: Blazing semantic trails between web resources. In: IEEE Internet Computing, 11(4), pp. 77–81 (2007)

APPENDIX A

Table 2. List of triggers grouped by their semantic relations

Semantic relations	**Lemmas of triggers (#occurrences)**	**Semantic relations**	**Triggers**
AFFECT.fear-of	*teme* (9), *frică* (3), *fior de groază* (1), *teamă* (1), *tremura* (1)	SOCIAL.colleague-of	*tovarăș* (8), *frate* (4), *aliat* (1), *compatrioată* (1), *coreligionar* (1), *oaspăt* (1), *protector* (1)
AFFECT.friend-of	*prieten/ prietenă/ prietenie* (28), *amic* (2), *pare rău* (1),	SOCIAL.in-competition-with	*adversar* (3), *întrece* (2), *bănuială* (1), *concura* (1), *favorit* (1), *rivală* (1)
AFFECT.hate	*ura/ ură/ urî* (10), *a nu iubi* (4), *disprețui* (1), *dușmănos* (1), *pârî* (1), *răzbuna* (1)	SOCIAL.in-cooperation-with	*ajutor* (1), *tovarăș* (1)
AFFECT.hated-by	*ură*, *urî* (3)		
AFFECT.love	*iubi* (44), *dragoste* (8), *adora* (7), *drag* (5), *iubit*/*iubită* (5), *îndrăgosti/ îndrăgostit* (5), *dor* (2), *favorit* (2), *afecțiune* (1), *devotat* (1), *îndurător* (1), *plăcea* (1), *scump* (1), *tânji* (1)	SOCIAL.inferior-of	*sclav/ sclavă* (33), *poruncă/ porunci* (19), *libert/ libertă* (14), *om* (14), *serv* (9), *adept* (7), *slugă* (6), *apostol* (5), *centurion* (4), *slujitor* (4), *condus* (3), *credincios* (3), *curtean* (3), *ostatecă* (3), *supune/ supunere/ supus* (3), *ales* (2), *ostaș* (2),
AFFECT.loved-by	*iubi* (36), *dragoste* (6), *drag* (5), *favorit* (2), *îndrăgi* (2), *admira* (1), *îndrăgostit* (1),		

	vrea iubire (1)		*preot* (2), *suit* (2), *arendaș* (1), *asculta* (1), *cohortă* (1), *consul* (1), *discipol* (1), *gardă* (1), *îngenunchea* (1), *învățăcel* (1), *învins* (1), *legiune* (1), *locțiitor* (1), *ordin* (1), *prosterna* (1), *rob* (1), *servitor* (1), *sluji* (1), *soldat* (1), *sub comandă* (1), *ucenic* (1), *un demn prozelit al lui Christos* (1), *servitor* (1), *victima* (1)
`AFFECT.rec-love`	*dragoste* (2), *amantă* (1), *îndrăgostită* (1), *iubi* (1)		
`AFFECT.upset`	*întrista* (1), *părea rău* (1)		
`AFFECT.worship`	*închina* (2), *ruga* (2), *cinsti* (1), *credincios* (1), *prosterna* (1), *slăvi* (1)		
`AFFECT.worshiped-by`	*adorată* (1), *adoratoare* (1), *slăvi* (1)		
`KINSHIP.aunt/uncle-of`	*unchi* (1)	`SOCIAL.opposite-to`	*dușman* (14), *împotrivi* (3), *împotriva* (2), *advers* (1), *bogat* (1), *călău* (1), *deosebit* (1), *învins* (1), *stăpân* (1)
`KINSHIP.child-of`	*fiu/ fiică* (69), *copil/ copilă* (21), *copie* (5), *băiat* (2), *vlăstar* (2), *fată* (1), *nepot* (1), *urmaș* (1)		
`KINSHIP.nephew-of`	*nepot* (8)	`SOCIAL.superior-of`	*stăpân/ stăpână* (27), *rege* (8), *mai mare/ mai marele/ mai marii* (7), *comandant* (6), *în frunte* (4), *comanda* (3), *prefect/ prefectură* (3), *conducere* (2), *învățător* (2), *mare* (2), *zeu* (2), *cârmuiește* (1), *conduce* (1), *dispune* (1), *domina* (1), *domn* (1), *dumnezeu* (1), *împărat* (1), *imperator* (1), *maestru* (1), *păzitor* (1), *sclav* (1), *superior* (1), *supraveghetor* (1)
`KINSHIP.parent-of`	*mamă* (19), *tată* (10), *părinte/ părinte adoptiv* (6), *fiu* (1)		
`KINSHIP.sibling`	*frate* (10), *soră* (8), *soție* (1)		
`KINSHIP.spouse-of`	*soție/ soț* (27), *logodnic/ logodnică* (6), *amantă* (4), *nevastă* (4), *iubită* (1)		
`KINSHIP.unknown`	*rudă* (9), *concubină* (2), *strămoș* (2), *neam* (1), *strănepot* (1), *urmaș* (1)		