A hybrid approach for relation extraction aimed to semantic annotations

Lucia Specia and Enrico Motta

Knowledge Media Institute & Centre for Research in Computing
The Open University - Walton Hall, MK7 6AA, Milton Keynes, UK
{L.Specia,E.Motta}@open.ac.uk

Abstract. We present an approach for relation extraction from texts aimed to enrich the semantic annotations produced by a semantic web portal. The approach exploits linguistic and empirical strategies, by means of a pipeline method involving processes such as a parser, part-of-speech tagger, named entity recognition system, pattern-based classification and word sense disambiguation models, and resources such as an ontology, knowledge base and lexical databases. With the use of knowledge intensive strategies to process the input data and corpus-based techniques to deal both with unpredicted cases and ambiguity problems, we expect to accurately discover most of the relevant relations for known and new entities, in an automated way.

1 Introduction

Relation Extraction (RE) consists of the identification of the semantic relations between pairs of terms in unstructured or semi-structured natural language documents. Semantic relations are useful for several applications, including the acquisition of terminological data, construction and extension of lexical resources and ontologies, question answering, information retrieval, semantic web annotation, etc.

In this paper we focus on the application of relation extraction to semantically annotate knowledge coming from raw text, as part of a framework aiming to automatically acquire high quality semantic metadata for the Semantic Web. One of the applications developed within this framework is the *KMi Semantic Web Portal*[1] [6], which analyzes data from texts, databases, and knowledge bases, in order to extract semantic knowledge from all of them in an integrated way, also verifying the quality of this knowledge, according to a domain ontology. The extracted knowledge is formalized into OCML and OWL representations[2].

Currently, the knowledge extracted by the semantic web portal from texts comprises mainly occurrences of entities (instances) that already exist in the knowledge base, and their properties also available in that knowledge base or in databases. It also includes occur-

[1] http://semanticweb.kmi.open.ac.uk:8080/ksw/index.html

[2] Examples of annotations produced by the KMi Semantic Web Portal for newsletters texts are available in http://plainmoor.open.ac.uk:8080/ksw/pages/news.jsp.

rences of new entities, as given by a named entity recognition system, according to the possible types of entities in the domain ontology. Thus, already existent entities are semantically annotated with their properties provided by the knowledge base and databases. However, new knowledge about entities (especially relational) is not taken into account. Moreover, little is done with new entities, which are annotated only with their types.

In that context, the relation extraction approach presented here aims to identify the semantic relations between entities in the input texts. These include already existent relations between the entities in the knowledge base, new relations predicted as possible by the domain ontology, or completely new (unpredicted) relations. Additionally, new entities are identified in a more comprehensive way, and their relations are also extracted. As a consequence, extra knowledge about (existing and new) entities can be acquired, yielding a richer representation of the input data, and helping to solve problems that arise when mapping this unstructured data into a semantic representation, such as ambiguities. By identifying new entities in the text and recognizing their types, the approach could also be applied to ontology population. Moreover, since it extracts new relations between entities, it could be used as a first step for ontology learning.

The relation extraction approach makes use of a domain ontology, a knowledge base, and lexical databases, along with knowledge-based and empirical resources and strategies for linguistic processing. These include a lemmatizer, syntactic parser, part-of-speech tagger, named entity recognition system, and pattern matching and word sense disambiguation models. The input data used in the experiments with our approach consists of English texts from the Knowledge Media Institute (KMi)[3] newsletters. We believe that by integrating corpus and knowledge-based techniques and using rich linguistic processing strategies in a completely automated and unsupervised fashion, the approach can achieve more effective results than the previous work, in terms of both accuracy and coverage.

In the remaining of this paper we first describe some cognate work on relation extraction, particularly those exploring empirical methods, for various applications (Section 2). We then present our approach, showing its architecture and describing each of its main components (Section 3). Finally, we discuss next steps (Section 4).

2 Related Work

Several approaches have been proposed for the extraction of relations from unstructured sources. Recently, they have focused on the use of supervised or unsupervised corpus-based techniques in order to automate the task. A very common approach is based on pattern matching, with patterns composed by subject-verb-object (SVO) tuples. Interesting work has been done on the unsupervised automatic definition of patterns from a small number of seed patterns. These are used as a starting point to bootstrap the pattern learning process, by means of semantic similarity measures [20, 16].

Most of the approaches for relation extraction rely on the mapping of syntactic dependencies, such as SVO, onto semantic relations, using either pattern matching or other strategies, such as probabilistic parsing for trees augmented with annotations for entities and relations [11], or clustering of semantically similar syntactic dependencies, according to

[3] http://kmi.open.ac.uk/

their selectional restrictions [5].

In corpus-based approaches, many variations are found concerning the machine learning techniques used to produce classifiers to judge relation as relevant or non-relevant. [14], e.g., uses probabilistic classifiers with constraints induced between relations and entities, such as selectional restrictions. Based on instances represented by a pair of entities and their position in a shallow parse tree, [17] uses support vector machines and voted perceptron as algorithms with a specialized kernel model. Also using kernel methods and support vector machines, [18] combines clues from different levels of syntactic information and applies composite kernels to integrate and extend the individual kernels.

The framework proposed by [6], still under development, similarly to our work aims at the automation of semantic annotations according to ontologies. Several supervised algorithms can be used on the training data represented through a canonical graph-based data model. The framework includes a shallow linguistic processing step, in which corpora are analyzed and a representation is produced according to the data model, and a classification step, where classifiers run on the datasets produced by the linguistic processing step.

Many relation extraction approaches have been also proposed focusing on the particular task of ontology development [10, 13, 15, 1]. These approaches aim to learn non-taxonomic relations between concepts, instead of lexical entries, addressed by traditional approaches within Information Extraction. However, in essence, they employ similar techniques, derived from text mining, to extract relations.

In the next section we describe our approach, which merges features that have shown to be effective in several of the previous works, in order to achieve more comprehensive and accurate results, aiming particularly at the generation of semantic annotation for the Semantic Web.

3 A hybrid approach for relation extraction

The proposed approach for relation extraction is illustrated in Fig. 1. It employs knowledge-based and (supervised and unsupervised) corpus-based techniques. The core strategy consists of mapping linguistic components with some syntactic relationship (a linguistic triple) into their corresponding semantic components. This includes mapping not only the relations, but also the linguistic terms linked by those relations. The identification of the linguistic triples involves a series of linguistic processing steps. The mapping between terms and concepts is guided by a domain ontology and a named entity recognition system. The identification of the relations relies on the knowledge available in the domain ontology and in a lexical database, and on pattern-based classification and sense disambiguation models.

The main goal of this approach is to provide rich semantic annotations that can be used, for example, by a semantic web portal. Since the resultant annotations include already existent and new entities and relations, there are other possible uses of our approach, including:

1) Ontology population: we map terms into new instances of concepts of an ontology and identify the relations between them, according to the possible relations in that ontology.

3) Ontology learning: we identify new relations between existent concepts, which can be used as a first step to extend an existent ontology. Certainly, a subsequent step to lift relations between instances to an adequate level of abstraction would be necessary (e.g., [10]).

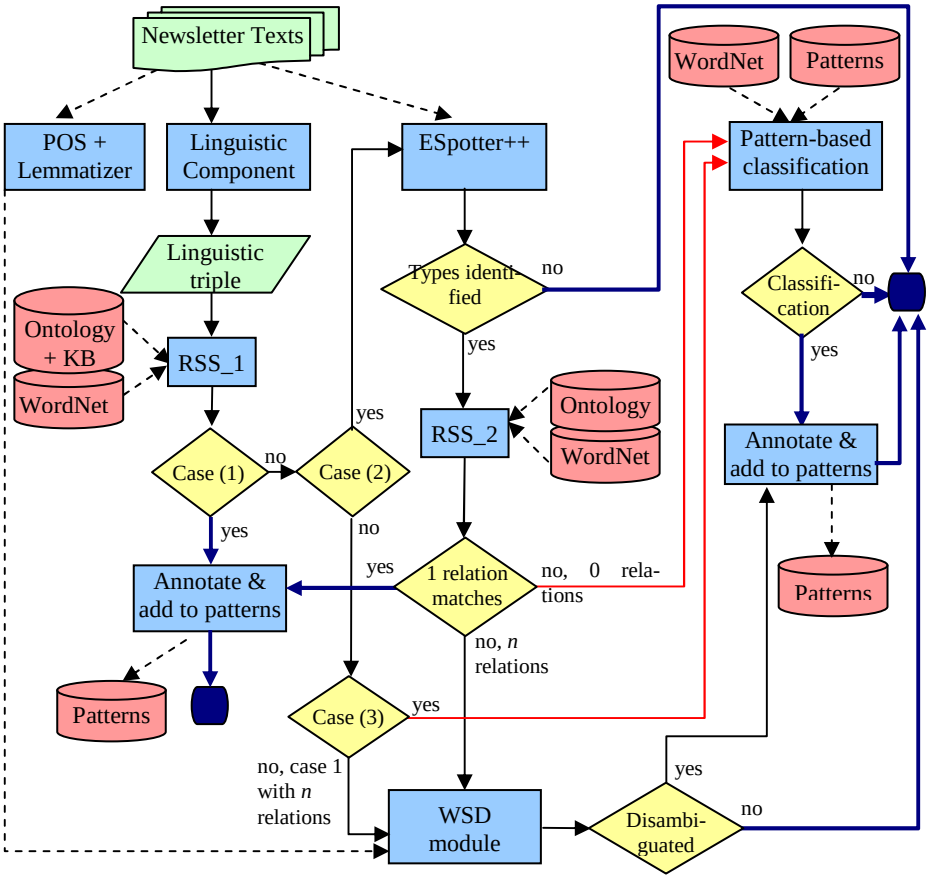

Fig. 1. Architecture of the proposed approach

3.1 Context and resources

The input to our experiments consists of electronic **Newsletter Texts** (KMi Planet[4]). These are short texts describing news of several natures related to KMi members: projects, publications, events, awards, etc. The domain **Ontology** used is the *KMi-basic-portal-ontology*. This was designed based on the AKT reference ontology[5] to include concepts relevant to the KMi domain. The instantiations of concepts in this ontology are stored in the knowledge base (**KB**) *KMi-basic-portal-kb*. The other two resources used in our architecture are the lexical database **WordNet** [4] and a repository of **Patterns** of relations, described in Section 3.4.

[4] http://news.kmi.open.ac.uk/kmiplanet/
[5] http://kmi.open.ac.uk/projects/akt/ref-onto/

3.2 Identifying linguistic triples

Given a newsletter text, the first step of the relation extraction approach is to process the natural language text in order to identify linguistic triples, that is, sets of three elements with a syntactic relationship, which can indicate potentially relevant semantic relations. In our architecture, this is accomplished by the **Linguistic Component** module. Part of this module is based on an adaptation of the linguistic component designed in Aqualog [9], a question answering system.

The linguistic component uses the infrastructure and the following resources from GATE [2]: tokenizer, sentence splitter, part-of-speech tagger, morphological analyzer and VP chunker. On the top of these resources, which produce syntactic annotations for the input text, the linguistic component uses a grammar to identify linguistic triples. This grammar was implemented in Jape [3], which allows the definition of patterns to recognize regular expressions using the annotations provided by GATE.

The main type of construction aimed to be identified by our grammar involves a verbal expression as indicative of a potential relation and two noun phrases as terms linked by that relation. However, our patterns also account for other types of constructions, including, e.g., the use of comma to implicitly indicate a relation, as in sentence (1). In this case, having identified that "KMi" is an *organization* and "Enrico Motta" is a *person*, it is possible to guess the relation indicated by the comma (for example, "work", resulting in the triple <enrico-motta, work, kmi>). Some examples triples identified by our patterns for the newsletter in Fig. 2 are given in Fig. 3.

(1) "Enrico Motta, at KMi now, is heading a project on ….".

Nobel Summit on ICT and public services

Peter Scott attended the Public Services Summit in Stockholm, during Nobel Week 2005. The theme this year was Responsive Citizen Centered Public Services. The event was hosted by the City of Stockholm and Cisco Systems Thursday 8 December - Sunday 11 December 2005.

The Nobel Week Summit provides an unusual venue to explore the possibilities of the Internet with top global decision-makers in education, healthcare and government and to honor the achievements of the 2005 Nobel Peace Prize Laureate.

Fig. 2. Example of newsletter

<peter-scott,attend,public-services-summit>
<public-services-summit,located,stockholm>
<theme,is,responsive-citizen-centered-public-services>
<city-of-stockholm-and-cisco-systems,host,event>
<the-nobel-week-summit,provide,unusual-venue>
<unusual-venue,explore,the-possibilities-of-the-internet>

Fig. 3. Examples of linguistic triples for the newsletter in Fig. 2

Although we were concerned about making the Jape patterns as comprehensive as possible, they are based on shallow syntactic information only, and therefore they are not able to capture certain potentially relevant triples. To overcome this limitation, we employ a parser as a complementary resource to produce linguistic triples. We use Minipar (Lin, 1993),

which produces functional relations for the components in a sentence, including subject and object relations with respect to a verb. This allows capturing some implicit relations, such as indirect objects and long distance dependence relations, which could not be identified by the Jape patterns. Fig. 4 shows some tuples extracted for the text in Fig. 2.

```
subject[peter_scott]+verb[attend]+verb_mod[during_nobel_week_2005]+
object[public_services_summit]+object_mod[in_stockholm]

subject[theme]+verb[be]+object[responsive]

subject[city]+subj_mod[of_stockholm]+verb[host]+object[event]
```

Fig. 4. Examples of tuples extracted from Minipar's dependency trees

Minipar's representation is converted into a triple format, repeating the verb when it is related to more than one subject or object. Thus, the intermediate representation provided both by GATE plus the Jape grammar and by Minipar consists of triples of the type: <noun_phrase, verbal_expression, noun_phrase>.

3.3 Identifying ontological entities and relations

Given a linguistic triple, the next step is to verify whether the verbal expression in that triple conveys a relevant semantic relationship between entities (given by the terms) potentially belonging to an ontology. This is the most important phase of our approach and is represented by a series of modules in our architecture in Fig. 1. As first step we try to map the linguistic triple into an ontology triple, by using an adaptation of the Relation Similarity Service (RSS) also developed in Aqualog [9].

RSS tries to make sense of the linguistic triple by looking at the structure of the domain ontology and the information stored in the KB. In order to map a linguistic triple into an ontology triple, besides looking for an exact matching between the components of the two triples, RSS considers partial matchings by using a set of resources in order to account for minor lexical or conceptual discrepancies between these two elements. These resources include metrics for string similarity matching, synonym relations given by WordNet, and a lexicon of previous mappings between the two types of triples.

RSS was originally designed to be used in an interactive fashion by a question answering system. Therefore, the user is expected to point out the appropriate mapping when there is no matching between the linguistic and ontology triples. The user is also expected to disambiguate among several options of mappings. In order to achieve a fully automated annotation process we use other modules to map linguistic triples into ontology triples even if there is no matching according to RSS (Section 3.4) or if there is ambiguity (Section 3.5).

Different strategies are employed to identify matchings for terms and relations, as explained in Sections 3.3.1 and 3.3.2. The application of these strategies to map the linguistic triples into existent or new instances and relations is described in Section 3.3.3.

3.3.1 Mapping terms

To map terms into entities, the following attempts are accomplished (in the given order):

1) Search the KB for an exact matching of the term with any instance.

2) Apply string similarity metrics[6] to calculate the similarity between the given term and each instance of the KB. A hybrid scheme combining three metrics is used: jaro-Winkler, jlevelDistance a wlevelDistance. It checks different combinations of threshold values for the metrics. The elements in our linguistic triples are lemmatized in order to avoid problems which could be incorrectly handled by the string similarity metrics (e.g., past tense).

2.1) If there is more that one possible matching, check whether any of them is a substring of the term. For example, the instance name for "Enrico Motta" is a substring of the term "Motta", and thus it should be preferred to any other instance.

For example, the similarity values returned for the term "vanessa" with instances potentially relevant for the mapping are given in Fig. 5. The combination of thresholds specified for the metrics is met for the instance "Vanessa Lopez", and thus the mapping is (correctly) accomplished. If there is still more than one possible mapping for a term in the linguistic triple, we assume there is not enough evidence to map that term, and the triple is discarded.

```
jaroDistance for "vanessa" and "vanessa-lopez" = 0.8461538461538461
wlevel for "vanessa" and "vanessa-lopez" = 1.0
jWinklerDistance for "vanessa" and "vanessa-lopez" = 0.9076923076923077
```

Fig. 5. String similarity measures for the term "vanessa" and the instance "Vanessa Lopez"

3.3.2 Mapping relations

In order to map the verbal expression into a conceptual relation, we assume that the terms of the triple have already been mapped either into instances of classes in the KB by RSS, or into potential new instances, by a named entity recognition system, as we explain will later in Section 3.3.3. The following attempts are then made for the verb-relation mapping:

1) Search the KB for an exact matching of the verbal expression with any existent relation for the instances under consideration or any possible relation between the classes (and superclasses) of the instances under consideration.

2) Apply the string similarity metrics to calculate the similarity between the given verbal expression and the possible relations between instances (or their classes) corresponding to the terms in the linguistic triple.

3) Search for similar mappings for the types/classes of entities under consideration in a lexicon of mappings previously accomplished according to users' choices in Aqualog[7]. This lexicon contains ontology triples along with the given verbal expression which was mapped to the conceptual relation, as illustrated in Table 1. The use of this lexicon represents a simplified form of pattern matching in which only exact matching is considered.

Table 1. Examples of lexicon patterns

given_relation	class_1	conceptual relation	class_2
works	project	has-project-member	person
cite	project	has-publication	publication

[6] Available in http://sourceforge.net/projects/simmetrics/.

[7] http://plainmoor.open.ac.uk:8080/aqualog/index.html

4) Search for synonyms of the given verbal expression in WordNet, in order to verify if there is a synonym that matches (complete or partially, using string similarity metrics) any existent relation for the instances under consideration, or any possible relation between the classes (or superclasses) of those instances (likewise in step 1).

If there is no possible mapping for the term, the pattern-based classification model is triggered (Section 3.4). Conversely, if there is more than one possible mapping, the disambiguation model is called (Section 3.5).

3.3.3 RSS for existing / new instances, and existent / predicted relations

In our architecture, RSS is represented by modules **RSS_1** and **RSS_2**. **RSS_1** first checks if the terms in the linguistic triple are instances of a KB (cf. described in Section 3.3.1). If the terms can be mapped to instances, it checks whether the relation given in the triple matches any already existent relation between for those instances, or, alternatively, if that relation matches any of the possible relations for the classes (and superclasses) of the two instances in the domain ontology (cf. Section 3.3.2). Three situations may arise from this attempt to map the linguistic triple into an ontology triple (Cases (1), (2), and (3) in Fig. 1):

Case (1) complete matching with instances of the KB and a relation of the KB or ontology, with possibly more than one valid conceptual relation being identified:

$$<\text{instance}_1, (\text{conceptual\_relation})^+, \text{instance}_2>.$$

Case (2) no matching or partial matching with instances of the ontology (the relation is not analyzed (*na*) when there is not a matching for instances):

$$<\text{instance}_1, na\ , ?> \quad \text{or} \quad <?, na, \text{instance}_2> \quad \text{or} \quad <?, na, ?>$$

Case (3) matching with instances of the KB, but no matching with a relation of the KB or ontology:

$$<\text{instance}_1, ?, \text{instance}_2>$$

If the matching attempt results in Case (1) with only one conceptual relation, then the triple can be formalized into a semantic annotation. This yields the annotation of an already existent relation for two instances of the KB, as well as a new relation for two instances of the KB, although this relation was already predicted in the ontology as possible between the classes of those instances. The generalization of the produced triple for classes/types of entities, i.e., <class, conceptual_relation, class>, is added to the repository of **Patterns**.

On the other hand, if there is more than one possible conceptual relation in case (1), the system tries to find the correct one by means of a sense disambiguation model, described in Section 3.5. Conversely, if there is no matching for the relation (Case (3)), the system tries an alternative strategy: the pattern-based classification model (Section 3.4). Finally, if there is no complete matching of the terms with instances of the KB (Case (2)), it means that the entities can be new to the KB.

In order to check if the terms in the linguistic triple express new entities, the system first identifies to what classes of the ontology they belong. This is accomplished by means of ESpotter++, and extension of the named entity recognition system ESpotter [19].

ESpotter is based on a mixture of lexicon (gazetteers) and patterns. We extended ESpotter by including new entities (extracted from other gazetteers), a few relevant new types of

entities, and a small set of efficient patterns. In Espotter++ all types of entities correspond to generic classes of our domain ontology. These types include: person, organization, event, publication, location, project, research-area, technology, date, etc.

In our architecture, if ESpotter++ is not able to identify the types of the entities, the process is aborted and no annotation is produced. This may be either because the terms do not have any conceptual mapping (for example "it"), or because the conceptual mapping is not part of our domain ontology. Otherwise, if ESpotter++ succeeds, we use the RSS again (**RSS_2**) in order to verify whether the verbal expression encompasses a semantic relation. Since at least one of the two entities is recognized by Espotter++, and therefore at least one entity is new, it is only possible to check if the relation matches one of the possible relations between the classes of the recognized entities (cf. Section 3.3.2).

If the matching attempt results in only one conceptual relation, then the triple will be formalized into a semantic annotation. This represents the annotation of a new (although predicted) relation and two or at least one new entity/instance. The produced triple of the type <class, conceptual_relation, class> is added to the repository of **Patterns**.

Again, if there are multiple valid conceptual relations, the system tries to find the correct one by means of a disambiguation model (Section 3.5). Conversely, if it there is no matching for the relation, the pattern-based classification model is triggered (Section 3.4).

3.4 Identifying new relations – the pattern matching model

The process described in Section 3.3 for the identification of relations accounts only for the relations already predicted as possible in the domain ontology. However, we are also interested in the additional information that can be provided by the text, in the form of new types of relations for known or new entities. In order to discover these relations, we employ a pattern matching strategy to identify relevant relations between types of terms.

The pattern matching strategy has proved to be an efficient way to extract semantic relations, but in general has the drawback of requiring the possible relations to be previously defined. In order to overcome this limitation, we employ a **Pattern-based classification** model that can identify similar patterns based on a very small initial number of patterns.

We consider patterns of relations between types of entities, instead of the entities themselves, since we believe that it would be impossible to accurately judge the similarity for the kinds of entities we are addressing (names of people, locations, etc). Thus, our patterns consist of triples of the type <class, conceptual_relation, class>, which are contrasted against a given triple using the classes already provided by the linguistic component or by ESpotter++ in order to classify relations in that triple as *relevant* or *non-relevant*.

The pattern-based classification model is based on the approach presented in [16]. It is an unsupervised corpus-based module which takes as examples a small set of relevant SVO patterns, called seed patterns, and uses a WordNet-based semantic similarity measure to compare the pattern to be classified against the relevant ones. Our initial seed patterns (see examples in Table 2) mixes patterns extracted from the lexicon generated by Aqualog's users (cf. Section 3.3.2) and a small number of manually defined relevant patterns. This set of patterns is expected to be enriched with new patterns as our system annotates relevant relations, since the system adds new triples to the initial set of patterns.

Table 2. Examples of seed patterns

class_1	conceptual relation	class_2
project	has-project-member	person
project	has-publication	publication
person	develop	technology
person	attend	event

Likewise [16], we use a semantic similarity metric based on the information content of the words in WordNet hierarchy, derived from corpus probabilities. It scores the similarity between two patterns by computing the similarity for each pair of words in those patterns. A threshold of 0.90 for this score was used here to classify two patterns as similar. In that case, a new annotation is produced for the input triple and it is added to the set of patterns.

It is important to notice that, although WordNet is also used in the RSS modules, in that case only synonyms are checked, while here the similarity metric explores deeper information in WordNet, considering the meaning (senses) of the words. It is also important to distinguish the semantic similarity metrics employed here from the string metrics used in RSS. String similarity metrics simply try to capture minor variations on the strings representing terms/relations, they do not account for the meaning of those strings.

3.5 Disambiguating relations

The ambiguity arising when more than one possible relation exists for a pair of entities is a problem neglected in most of the current work on relation extraction. In our architecture, when the RSS finds more than one possible relation, we try to choose one relation by using the word sense disambiguation (**WSD**) system SenseLearner [12].

SenseLearner is minimally supervised WSD system to disambiguate all open class words in any given text, after being trained on a small data set, according to global models for word categories. The current distribution includes two default models for verbs, which were trained on a corpus containing 200,000 content words of journalistic texts manually tagged with their WordNet senses. Since SenseLeaner requires a corpus tagged with senses in order to be trained to specific domains and there is not such a corpus for our domain, we use one of the default training models, which accounts for the most common uses of the verbs. This is a contextual model that relies on the first word before and after the verb, and its POS tags. To disambiguate new cases, it requires only that these cases are annotated with the POS tags of the words. The use of lemmas of the words instead of the words yields better results, since the models were generated for lemmas. In our architecture, the POS and lemma annotations are produced by the component **POS + Lemmatizer.**

Since the WSD module disambiguates among WordNet senses, it is employed only after the use of the WordNet subcomponent by RSS. This subcomponent finds all the synonyms for the verb in a linguistic triple and checks which of them matches existent or possible relations for the terms in that triple. In some cases, however, there is a matching for more than one synonym. In WordNet, synonyms usually represent different uses of the verb. Therefore, the WSD module is used to identify in which sense the verb is being used in the sentence, allowing the system to choose one among the possible matchings.

For example, given the linguistic triple <enrico_motta, head, kmi>, RSS is able to identify that “enrico_motta” is a *person*, and that “kmi” is an *organization*. However, it cannot

find an exact or partial matching (using string metrics), or even a matching given by the user lexicon. After getting the synonyms for "head" in WordNet, RSS verifies that two of them match possible relations in the ontology between a *person* and an *organization*: "direct" and "lead". In this case, the WSD module correctly disambiguates the sense of "head" in the input sentence from which the linguistic triple was produced as "direct".

3.6 Annotating relevant relations

To formalize the relations extracted, we use the representation specified for the KMi Semantic Web Portal, in order to make it straightforward to integrate this knowledge to the one produced by the portal. The representation of the entity "Enrico Motta" and of all the relations involving this entity from the news text in Fig. 6, e.g., is given in Fig. 7.

KMi awarded £4M for Semantic Web Research

Professor Enrico Motta and Dr John Domingue of the Knowledge Media Institute have received a set of record-breaking awards totalling £4m from the European Commission's Framework 6 Information Society Technologies (IST) programme. This is the largest ever combined award obtained by KMi associated with a single funding programme. The awards include three Integrated Projects (IPs) and three Specific Targeted Research Projects (STREPs) and they consolidate KMi's position as one of the leading international research centers in semantic technologies. Specifically Professor Motta has been awarded:

a.. £1.55M for the project NeOn: Lifecycle Support for Networked Ontologies
b.. £565K for XMEDIA: Knowledge Sharing and Reuse across Media and
c.. £391K for OK: Openknowledge - Open, coordinated knowledge sharing architecture. …

Fig. 6. Example of newsletter

```
(def-instance Enrico-Motta kmi-academic-staff-member
 ((works-in knowledge-media-institute)
  (award-from european-commission)
  (award-for NeOn)
  (award-for XMEDIA)
  (award-for OK)))
```

Fig. 7. Semantic annotations produced for the news in Fig. 6

In this case, "Enrico-Motta" is an instance of *kmi-academic-staff-member*, a subclass of *person* in the domain ontology. The mapped relation "works-in" "knowledge-media-institute" already existed in the KB. The new relations pointed out by our approach are the ones referring to the award received from the "European Commission" (an *organization*, here), for three *projects*: "NeOn", "XMEDIA", and "OK".

4 Conclusions and future work

We presented a hybrid approach for the extraction of semantic relations from text. It was designed mainly to enrich the annotations produced by a semantic web portal, but can be used for other domains and applications, such as ontology population and development.

Currently we are concluding the integration of the several modules composing our architecture. We will then carry experiments with our corpus of newsletters in order to evaluate the approach. Subsequently, we will incorporate the architecture to the semantic web portal and accomplish an extrinsic evaluation in the context of that application. Since the approach uses deep linguistic processing and corpus-based strategies not requiring any manual annotation, we expect it will accurately discover most of the relevant relations in the text.

References

1. Ciaramita, M., Gangemi, A., Ratsch, E., Saric, J., Rojas, I. Unsupervised learning of semantic relations between concepts of a molecular biology ontology. 19th IJCAI (2005) 659-664
2. Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. 40th ACL Meeting, Philadelphia (2002)
3. Cunningham, H., Maynard, D., and Tablan, V. JAPE: a Java Annotation Patterns Engine. Tech. Report CS--00--10, University of Sheffield, Department of Computer Science (2000)
4. Fellbaum, C. D. (ed). Wordnet: An Electronic Lexical Database. The MIT Press (1998)
5. Gamallo, P., Gonzalez, M., Agustini, A., Lopes, G., de Lima, V.S. Mapping syntactic dependencies onto semantic relations. ECAI Workshop on Machine Learning and Natural Language Processing for Ontology Engineering, Lyon, France (2002)
6. Iria, J. and Ciravegna, F. Relation Extraction for Mining the Semantic Web. Dagstuhl Seminar on Machine Learning for the Semantic Web, Dagstuhl, Germany (2005)
7. Lei, Y., Sabou, M., Lopez, V., Zhu, J., Uren, V., and Motta, E. An infrastructure for Acquiring High Quality Semantic Metadata. To appear in the 3rd ESWC, Budva, Montenegro (2006)
8. Lin, D. Principle based parsing without overgeneration. 31st ACL, Columbus (1993) 112-120
9. Lopez, V., Pasin, M., and Motta, E. AquaLog: An Ontology-portable Question Answering System for the Semantic Web. ESWC 2005, Creete, Grece (2005)
10. Maedche, A., Staab, S. Ontology learning for the semantic web. IEEE Intelligent Systems 16 (2001) 72-79
11. Miller, S., Fox, H., Ramshaw, L.A. and Weischedel, R.M. A novel use of statistical parsing to extract information from text. 6th ANLP-NAACL, Seattle (2000) 226-233
12. Mihalcea, R. and Csomai, A. SenseLearner: Word Sense Disambiguation for All Words in Unrestricted Text. 43rd ACL Meeting, Ann Arbor (2005)
13. Reinberger, M.L., Spyns, P. Discovering knowledge in texts for the learning of DOGMA inspired ontologies. ECAI 2004 Workshop on Ontology Learning and Population, Valencia (2004) 19-24
14. Roth, D., Yih, W. T. Probabilistic reasoning for entity & relation recognition. 19th COLING, Taipei, Taiwan (2002) 1-7
15. Schutz, A. and Buitelaar, P. RelExt: A Tool for Relation Extraction from Text in Ontology Extension. 4th ISWC (2005) 593-606
16. Stevenson, M. An Unsupervised WordNet-based Algorithm for Relation Extraction. 4th LREC Workshop Beyond Named Entity: Semantic Labeling for NLP Tasks, Lisbon (2004).
17. Zelenko, D., Aone, C., and Richardella, A. Kernel Methods for Relation Extraction. Journal of Machine Learning Research (2003). 3:1083-1106.
18. Zhao, S., Grishman, R. Extracting Relations with Integrated Information Using Kernel Methods. 43d ACL Meeting, Ann Arbor (2005)
19. Zhu, J., Uren, V., and Motta, E. ESpotter: Adaptive Named Entity Recognition for Web Browsing. 3rd Conference on Professional Knowledge Management, Kaiserslautern (2005) 518-529
20. Yangarber, R., Grishman, R., Tapanainen, P. Unsupervised Discovery of Scenario-Level Patterns for Information Extraction. 6th ANLP (2000) 282-289