2006

COLING • ACL

COLING • ACL 2006

2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge

Proceedings of the Workshop

Chairs:
Paul Buitelaar, Philipp Cimiano and Berenike Loos

22 July 2006
Sydney, Australia

Production and Manufacturing by
BPA Digital
11 Evans St
Burwood VIC 3125
AUSTRALIA

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
`acl@aclweb.org`

ISBN 1-932432-77-9

Table of Contents

Preface

An ontology is an explicit and formal specification of a shared conceptualization of a domain of interest. Ontologies formalize the intensional aspects of a domain, whereas the extensional part is provided by a knowledge base that contains assertions about instances of concepts and relations as defined by the ontology. The process of defining and instantiating a knowledge base is referred to as knowledge mark-up or ontology population, whereas (semi-)automatic support in ontology development is usually referred to as ontology learning.

Ontologies have been broadly used in knowledge management applications, including Semantic Web applications and research. In recent years, ontologies have regained interest also within the NLP community, specifically in such applications as information extraction, text mining and question answering. However, as ontology development is a tedious and costly process there has been an equally growing interest in the automatic learning of ontologies. Much of this work has been focused on textual data as human language is a primary mode of knowledge transfer. In this way, textual data provide both a resource for the ontology learning process as well as an application medium for developed ontologies.

Automatic methods for text-based ontology learning and population have developed over recent years, but it is difficult to compare approaches and results. In the 1st Workshop on Ontology Learning and Population (at ECAI 2004) we addressed this issue through an emphasis on the evaluation aspects of the reported work. The proceedings of this second workshop on ontology learning and evaluation (OLP2) contain 8 high-quality peer-reviewed papers presenting novel approaches that address diverse topics within ontology learning, i.e. learning taxonomic and non-taxonomic relations, populating ontologies with named entities and instances of relations as well as lexical enrichment of ontologies. Thanks to the excellent work of the program committee we have been able to compile an interesting and high quality program.

The program is divided into three thematic parts: “Lexical Ontology Enrichment”, “Ontology Population and Ontology-based IE” and “Taxonomy and Relation Extraction”. The workshop will conclude with two invited talks by Dekang Lin and Johan Bos on the usefulness of ontology learning, leading to a hopefully vivid discussion. We hope you enjoy the workshop.

Paul Buitelaar, DFKI, Saarbrücken, Germany
Philipp Cimiano, AIFB Karlsruhe, Germany
Berenike Loos, European Media Laboratory, Heidelberg, Germany

Organizers

Chairs:

Paul Buitelaar, DFKI, Germany
Philipp Cimiano, AIFB, Univ. of Karlsruhe, Germany
Berenike Loos, European Media Lab, Germany

Program Committee:

Eneko Agirre, Basque Country University, Spain
Enrique Alfonseca, Universidad Autónoma de Madrid, Spain
Nathalie Aussenac-Gilles, IRIT- CNRS Toulouse, France
Timothy Baldwin, University of Melbourne, Australia
Roberto Basili, Università di Roma “Tor Vergata”, Italy
Johan Bos, Università di Roma “La Sapienza”, Italy
Christopher Brewster, University of Sheffield, UK
Massimiliano Ciaramita, LOA-ISTC, Italy
Nigel Collier, National Institute of Informatics, Japan
Ido Dagan, Bar Ilan University, Israel
Eric Gausier, XEROX XRCE, France
Asuncion Gomez-Perez, Universidad Politécnica de Madrid, Spain
Marko Grobelnik, Jožef Stefan Institute, Slovenia
Siegfried Handschuh, DERI Galway, Ireland
Andreas Hotho, University of Kassel, Germany
Eduard Hovy, USC, Information Sciences Institute, USA
Vipul Kashyap, Partners HealthCare System, USA
Bernardo Magnini, ITC-IRST, Italy
Diana Maynard, University of Sheffield, UK
Adeline Nazarenko, LIPN - Université Paris-Nord, France
Claire Nedellec, MIG, INRA, France
George Paliouras, NCSR “Demokritos”, Greece
Patrick Pantel, USC, Information Sciences Institute, USA
Robert Porzel, European Media Lab, Germany
Marie-Laure Reinberger, Universiteit Antwerpen, Belgium
Marta Sabou, Knowledge Media Institute, UK
Michael Sintek, DFKI, Germany
Peter Spyns, Vrije Universiteit Brussel, Belgium
Steffen Staab, University of Koblenz-Landau, Germany
Vojtech Svatek, University of Economics, Prague, Czech Rep.
Paola Velardi, Università di Roma “La Sapienza”, Italy
Dominic Widdows, MAYA Design, USA

Invited Speakers:

Johan Bos, Università di Roma “La Sapienza”, Italy
Dekang Lin, University of Alberta, Canada

Workshop Program

Saturday, 22 July 2006

9:00–9:30 Introduction to OLP and Overview of the Workshop

Session 1: Lexical Ontology Enrichment

9:30–10:00 *Enriching a Formal Ontology with a Thesaurus: an Application in the Cultural Heritage Domain*
Roberto Navigli and Paola Velardi

10:00–10:30 *Multilingual Ontology Acquisition from Multiple MRDs*
Eric Nichols, Francis Bond, Takaaki Tanaka, Sanae Fujita and Dan Flickinger

10:30–11:00 Coffee Break

Session 2: Ontology Population and Ontology-Based IE

11:00–11:30 *LEILA: Learning to Extract Information by Linguistic Analysis*
Fabian M. Suchanek, Georgiana Ifrim and Gerhard Weikum

11:30–12:00 *Ontology Population from Textual Mentions: Task Definition and Benchmark*
Bernardo Magnini, Emanuele Pianta, Octavian Popescu and Manuela Speranza

12:00–12:30 *Efficient Hierarchical Entity Classifier Using Conditional Random Fields*
Koen Deschacht and Marie-Francine Moens

12:30–14:00 Lunch Break

Session 3: Taxonomy and Relation Extraction

14:00–14:30 *Taxonomy Learning using Term Specificity and Similarity*
Pum-Mo Ryu and Key-Sun Choi

14:30–15:00 *Towards Large-scale Non-taxonomic Relation Extraction: Estimating the Precision of Rote Extractors*
Enrique Alfonseca, Maria Ruiz-Casado, Manabu Okumura and Pablo Castells

15:00–15:30 *A hybrid approach for extracting semantic relations from texts*
Lucia Specia and Enrico Motta

15:30–16:00 Coffee Break

16:00–17:30 Invited Talks and Panel Discussion by Johan Bos and Dekang Lin

17:30–18:00 Concluding Remarks

Enriching a formal ontology with a thesaurus: an application in the cultural heritage domain

Roberto Navigli, Paola Velardi
Dipartimento di Informatica, Università “La Sapienza”, Italy
navigli,velardi@di.uniroma1.it

Abstract

This paper describes a pattern-based method to automatically enrich a core ontology with the definitions of a domain glossary. We show an application of our methodology to the cultural heritage domain, using the CIDOC CRM core ontology. To enrich the CIDOC, we use available resources such as the AAT art and architecture glossary, WordNet, the Dmoz taxonomy for named entities, and others.

1 Introduction

Large-scale, automatic semantic annotation of web documents based on well established domain ontologies would allow various Semantic Web applications to emerge and gain acceptance. Wide coverage ontologies are indeed available for general-purpose domains (e.g. WordNet, CYC, SUMO[1]), however semantic annotation in unconstrained areas seems still out of reach for state of art systems. Domain-specific ontologies are preferable since they limit the domain and make the applications feasible. Furthermore, real-world applications (e.g tourism, cultural heritage, e-commerce) are dominated by the requirements of the related web communities, who began to believe in the benefits deriving from the application of Semantic Web techniques. These communities are interested in extracting from texts specific types of information, rather than general-purpose relations. Accordingly, they produced remarkable efforts to conceptualize their competence domain through the definition of a *core ontology*[2]. Relevant examples are in the area of enterprise modeling (Fox et al. 1997) (Uschold et al. 1998) and cultural heritage (Doerr, 2003).

Core ontologies are indeed a necessary starting point to model in a principled way the basic concepts, relations and axioms of a given domain. But in order for an ontology to be really *usable* in applications, it is necessary to enrich the core structure with the thousands of concepts and instances that “make” the domain.

In this paper we present a methodology to automatically annotate a glossary $\mathcal{G}$ with the semantic relations of an existing *core ontology* $\mathcal{O}$. Glosses are then converted into formal concepts, used to enrich $\mathcal{O}$. The annotation of glossary definitions is performed using regular expressions, a widely adopted text mining approach. However, while in the literature regular expressions seek mostly for patterns at the lexical and part of speech level, we defined more complex expressions enriched with syntactic and semantic constraints. A word sense disambiguation algorithm, SSI (Velardi and Navigli, 2005), is used to automatically replace the high level semantic constraints specified in the core ontology with fine–grained sense restrictions, using the sense inventory of a general purpose lexicalized ontology, WordNet.

We experimented our methodology in the cultural heritage domain, since for this domain several well-established resources are available, like the CIDOC-CRM core ontology, the AAT art and architecture thesaurus, and others.

The paper is organized as follows: in Section 2 we briefly present the CIDOC and the other resources used in this work. In Section 3 we describe in detail the ontology enrichment algorithm. Finally, in Section 4 we provide a performance evaluation on a subset of CIDOC

[1] WordNet: http://wordnet.princeton.edu, CYC: http://www.opencyc.org, SUMO: http://www.ontologyportal.org

[2] a core ontology is a very basic ontology consisting only of the minimal concepts relations and axioms required to understand the other concepts in the domain.

Proceedings of the 2nd Workshop on Ontology Learning and Population, pages 1–9,
Sydney, July 2006. ©2006 Association for Computational Linguistics

properties and a sub-tree of the AAT thesaurus. Related literature is examined in Section 5.

2 Semantic and lexical resources in the cultural heritage domain

In this section we briefly describe the resources that have been used in this work.

2.1 The CIDOC CRM

The core ontology $\mathcal{O}$ is the *CIDOC CRM* (Doerr, 2003), a formal core ontology whose purpose is to facilitate the integration and exchange of cultural heritage information between heterogeneous sources. It is currently being elaborated to become an ISO standard. In the current version (4.0) the CIDOC includes 84 taxonomically structured concepts (called *entities*) and a flat set of 141 semantic relations, called *properties*. Properties are defined in terms of *domain* (the class for which a property is formally defined) and *range* (the class that comprises all potential values of a property), e.g.:

P46 is composed of (forms part of)
Domain: E19 Physical Object
Range: E42 Object Identifier

The CIDOC is an "informal" resource. To make it usable by a computer program, we replaced specifications written in natural language with formal ones. For each property $\mathcal{R}$, we created a tuple $R(C_d,C_r)$ where C_d and C_r are the domain and range entities specified in the CIDOC reference manual.

2.2 The AAT thesaurus

The domain glossary $\mathcal{G}$ is the Art and Architecture Thesaurus (AAT) a controlled vocabulary for use by indexers, catalogers, and other professionals concerned with information management in the fields of art and architecture. In its current version[3] it includes more than 133,000 terms, descriptions, bibliographic citations, and other information relating to fine art, architecture, decorative arts, archival materials, and material culture. An example is the following:

maestà
Note: Refers to a work of a specific iconographic type, depicting the Virgin Mary and Christ Child enthroned in the center with saints and angels in adoration to each side. The type developed in Italy in the 13th century and was based on earlier Greek types. Works of this type are typically two-dimensional, including painted panels (often altarpieces), manuscript illuminations, and low-relief carvings.
Hierarchical Position:
Objects Facet
.... Visual and Verbal Communication
.... Visual Works
.... <visual works>
.... <visual works by subject type>
.... maestà

We manually mapped the top CIDOC entities to AAT concepts, as shown in Table 1.

AAT topmost	CIDOC entities
Top concept of AAT	CRM Entity (E1), Persistent Item (E77)
Styles and Periods	Period (E4)
Events	Event (E5)
Activities Facet	Activity (E7)
Processes/Techniques	Beginning of Existence (E63)
Objects Facet	Physical Stuff (E18), Physical Object (E19)
Artifacts	Physical Man-Made Stuff (E24)
Materials Facet	Material (E57)
Agents Facet	Actor (E39)
Time	Time-Span (E52)
Place	Place (E53)

Table 1: mapping between AAT and CIDOC.

2.3 Additional resources

A general purpose lexicalised ontology, WordNet, is used to bridge the high level concepts defined in the core ontology with the words in a fragment of text. As better clarified later, WordNet is used to verify that certain words in a string of text *f* satisfy the range constraints $R(C_d,C_r)$ in the CIDOC. In order to do so, we manually linked the WordNet topmost concepts to the CIDOC entities. For example, the concept E19 (Physical Object) is mapped to the WordNet synset "*object, physical object*". Furthermore, we created a gazetteer $\mathcal{I}$ of named entities extracting names from the Dmoz[4], a large human-edited directory of the web, the Union List of Artist Names (ULAN) and the Getty Thesaurus of Geographic Names (GTG) provided by the Getty institute, along with the AAT. Named entities often occur in AAT definitions, therefore, NE recognition is relevant for our task.

[3] http://www.getty.edu/research/conducting_research/vocabularies/aat/

[4] http://dmoz.org/about.html

3 Enriching the CIDOC CRM with the AAT thesaurus

In this Section we describe in detail the method for automatic semantic annotation and ontology enrichment in the cultural heritage domain.

We start with an example of the task to be performed: given a gloss *G* of a term *t* in the glossary $\mathcal{G}$, the first objective is to *annotate* certain gloss fragments with CIDOC relations. For example, the following gloss fragment for "*vedute*" is annotated with a CIDOC relation, as follows:

[..]The first vedute probably were **<carried-out-by>***painted by northern European artists***</carried-out-by>** *[...]*

Then, for each annotated fragment, we extract a *semantic relation instance* R(C_t,C_w), where R is a relation in $\mathcal{O}$, C_t and C_w are respectively the *domain* and *range* of R. The concept C_t corresponds to its lexical realization *t*, while C_w is the concept associated to the "head" word *w* in the annotated segment of the gloss.

In the previous example, the relation instance is: $R_{\text{carried\_out\_by}}$(*vedute,European_artist*)

The annotation process allows to automatically enrich $\mathcal{O}$ with an existing glossary in the same domain of $\mathcal{O}$, since each pair of term and gloss (t,G) in the glossary $\mathcal{G}$ is transformed into a *formal definition*, compliant with $\mathcal{O}$. Furthermore, the very same method used to annotate definitions can be used to annotate *free text* with the relations of the enriched ontology $\mathcal{O}'$.

We now describe the method in detail. Let $\mathcal{G}$ be a glossary, *t* a term in $\mathcal{G}$ and *G* the corresponding natural language definition (gloss). The main steps of the algorithm are the following:

1. Part-of-Speech analysis.

Each input gloss is processed with a part-of-speech tagger, TreeTagger[5]. As a result, for each gloss $G = w_1\ w_2\ ...\ w_n$, a string of part-of-speech tags $p_1\ p_2\ ...\ p_n$ is produced, where $p_i \in \mathcal{P}$ is the part-of-speech tag chosen by TreeTagger for word w_i, and $\mathcal{P} = \{\ N, A, V, J, R, C, P, S, W\ \}$ is a simplified set of syntactic categories (respectively, nouns, articles, verbs, adjectives, adverbs, conjunctions, prepositions, symbols, wh-words). Terminological strings (european artist) are detected using our Term Extractor tool, already described in (Navigli and Velardi, 2004).

2. Named Entity recognition.

We augmented TreeTagger with the ability to capture named entities of locations, organizations, persons, numbers and time expressions. In order to do so, we use regular expressions (Friedl, 1997) in a rather standard way, therefore we omit details. When a named entity string $w_i\ w_{i+1}\ ...\ w_{i+j}$ is recognized, it is transformed into a single term and a specific part of speech denoting the kind of entity is assigned to it (*L* for cities (e.g. Venice), countries and continents, *T* for time and historical periods (e.g. Middle Ages), *O* for organizations and persons (e.g. Leonardo Da Vinci), B for numbers).

3. Annotation of sentence segments with CIDOC properties.

Once the text has been parsed, we use manually defined regular expressions to capture relevant fragments. The regular expressions are used to annotate gloss segments with properties grounded on the CIDOC-CRM relation model. Given a gloss *G* and a property[6] *R*, we define a *relation checker* c_R taking in input *G* and producing in output a set F_R of fragments of *G* annotated with the property *R*: <*R*>*f*</*R*>. The selection of a fragment *f* to be included in the set F_R is based on three different kinds of constraints:

- a **part-of-speech constraint** *p*(*f*, *pos-string*) matches the part-of-speech (*pos*) string associated with the fragment *f* against a regular expression (*pos-string*), specifying the required syntactic structure.
- a **lexical constraint** *l*(*f*, *k*, *lexical-constraint*) matches the lemma of the word in *k-th* position of *f* against a regular expression (*lexical-constraint*), constraining the lexical conformation of words occurring within the fragment *f*.
- **semantic constraints on domain and range** s_D(*f*, *semantic-domain*) and *s*(*f*, *k*, *semantic-range*) are valid, respectively, if the term *t* and the word in the *k-th* position of *f* match the semantic constraints on domain and range imposed by the CIDOC, i.e. if there exists at least one sense of *t* C_t and one sense of *w* C_w such that:

[5] TreeTagger is available at: http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger.

[6] In what follows, we adopt the CIDOC terminology for relations and concepts, i.e. properties and entities.

$R_{kind\text{-}of}*(C_d, C_t)$ and $R_{kind\text{-}of}*(C_r, C_w)$[7]

More formally, the annotation process is defined as follows:

A *relation checker* c_R for a property R is a logical expression composed with constraint predicates and logical connectives, using the following production rules:

$c_R \rightarrow s_D(f, \textit{semantic-domain}) \wedge c_R'$
$c_R' \rightarrow \neg c_R' \mid (c_R' \vee c_R') \mid (c_R' \wedge c_R')$
$c_R' \rightarrow p(f, \textit{pos-string}) \mid l(f, k, \textit{lexical-constraint})$
$\mid s(f, k, \textit{semantic-range})$

where f is a variable representing a sentence fragment. Notice that a relation checker must always specify a semantic constraint s_D on the *domain* of the relation R being checked on fragment f. Optionally, it must also satisfy a semantic constraint s on the k-th element of f, the range of R.

For example, the following excerpt of the checker for the *is-composed-of* relation (*P46*):

(1) $c_{is\text{-}composed\text{-}of}(f) = s_D(f, \textit{physical object\#1})$
$\wedge$ $p(f,$ "(V)$_1$(P)$_2$R?A?[CRJVN]*(N)$_3$")
$\wedge$ $l(f, 1,$
"^(consisting|composed|comprised|constructed)$")
$\wedge$ $l(f, 2,$ "of") $\wedge$ $s(f, 3, \textit{physical\_object\#1})$

reads as follows: "the fragment f is valid if it consists of a verb in the set { *consisting*, *composed*, *comprised*, *constructed* }, followed by a preposition "of", a possibly empty number of adverbs, adjectives, verbs and nouns, and terminated by a noun interpretable as a *physical object* in the WordNet concept inventory". The first predicate, s_D, requires that also the term t whose gloss contains f (i.e., its domain) be interpretable as a *physical object*.

Notice that some letter in the regular expression specified for the part-of-speech constraint is enclosed in parentheses. This allows it to identify the relative positions of words to be matched against lexical and semantic constraints, as shown graphically in Figure 1.

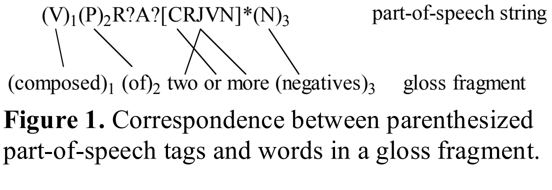


Figure 1. Correspondence between parenthesized part-of-speech tags and words in a gloss fragment.

Checker (1) recognizes, among others, the following fragments (the words whose part-of-speech tags are enclosed in parentheses are indicated in bold):

(consisting)$_1$ **(of)$_2$** semi-precious **(stones)$_3$** (matching part-of-speech string: **(V)$_1$(P)$_2$** J**(N)$_3$**)

(composed)$_1$ **(of)$_2$** **(knots)$_3$** (matching part-of-speech string: **(V)$_1$(P)$_2$(N)$_3$**)

As a second example, an excerpt of the checker for the *consists-of* (*P45*) relation is the following:

(2) $c_{consists\text{-}of}(f) = s_D(f, \textit{physical object\#1})$
$\wedge p(f,$ "(V)$_1$(P)$_2$A?[JN,VC]*(N)$_3$")
$\wedge$ $l(f, 1,$ "^(make|do|produce|decorated)$")
$\wedge$ $l(f, 2,$ "^(of|by|with)$")
$\wedge$ $\neg s(f, 3, \textit{color\#1}) \wedge \neg s(f, 3, \textit{activity\#1})$
$\wedge$ $(s(f, 3, \textit{material\#1}) \wedge s(f, 3, \textit{solid\#1})$
$\wedge$ $s(f, 3, \textit{liquid\#1}))$

recognizing, among others, the following phrases:

- **(made)$_1$** **(with)$_2$** the red earth pigment **(sinopia)$_3$** (matching part-of-speech string: **(V)$_1$(P)$_2$**AJNN**(N)$_3$**)
- **(decorated)$_1$** **(with)$_2$** red, black, and white **(paint)$_3$** (matching part-of-speech string: **(V)$_1$(P)$_2$**JJCJ**(N)$_3$**)

Notice that in both checkers (1) and (2) semantic constraints are specified in terms of WordNet sense numbers (*material#1*, *solid#1* and *liquid#1*), and can also be negative (*¬color#1* and *¬activity#1*). The motivation is that *CIDOC constraints are coarse-grained* due to the small number of available core concepts: for example, the property *P45 consists of* simply requires that the range belongs to the class *Material* (*E57*). Using these coarse grained constraints would produce *false positives* in the annotation task, as discussed later. Using WordNet for semantic constraints has two advantages: first, it is possible to write more fine-grained (and hence more reliable) constraints, second, regular expressions can be re-used, at least in part, for other domains and ontologies. In fact, several CIDOC properties are rather general-purpose.

Notice that, as remarked in section 2.3, replacing coarse CIDOC sense restrictions with WordNet fine-grained restrictions is possible since we mapped the 84 CIDOC entities onto WordNet topmost concepts.

4. Formalisation of glosses.

The annotations generated in the previous step are the basis for extracting *property instances* to enrich the CIDOC CRM with a conceptualization of the AAT terms. In general, for each gloss G defining a concept C_t,

[7] $R_{kind\text{-}of}*$ denotes zero, one, or more applications of $R_{kind\text{-}of}$.

and for each fragment $f \in F_R$ of G annotated with the property R: <*R*>*f*</*R*>, it is possible to extract one or more property instances in the form of a triple $R(C_t, C_w)$, where C_w is the *concept* associated with a term or multi-word expression *w* occurring in *f* (i.e. its language realization) and C_t is the *concept* associated to the defined term *t* in AAT. For example, from the definition of *tatting* (a kind of lace) the algorithm automatically annotates the phrase *composed of knots*, suggesting that this phrase specifies the *range* of the *is-composed-of* property for the term *tatting*:

$$R_{is\text{-}composed\text{-}of}(C_{tatting}, C_{knot})$$

In this property instance, $C_{tatting}$ is the *domain* of the property (a term in the AAT glossary) and C_{knot} is the *range* (a specific term in the definition *G* of *tatting*).

Selecting the concept associated to the domain is rather straightforward: glossary terms are in general not ambiguous, and, if they are, we simply use a numbering policy to identify the appropriate concept. In the example at hand, $C_{tatting}$=*tatting#1* (the first and only sense in AAT). Therefore, if C_t matches the domain restrictions in the regular expression for *R*, then the domain of the relation is considered to be C_t. Selecting the range of a relation is instead more complicated. The first problem is to select the correct words in a fragment *f*. Only certain words of an annotated gloss fragment can be exploited to extract the range of a property instance. For example, in the phrase "depiction of fruit, flowers, and other objects" (from the definition of *still life*), only *fruit*, *flowers*, *objects* represent the range of the property instances of kind *depicts* (*P62*).

When writing relation checkers, as described in the previous paragraph of this Section, we can add *markers of ontological relevance* by specifying a predicate *r*(*f*, *k*) for each relevant position *k* in a fragment *f*. The purpose of these markers is precisely to identify words in *f* whose corresponding concepts are in the range of a property. For instance, the checker (1) $c_{is\text{-}composed\text{-}of}$ from the previous paragraph is augmented with the conjunction: ∧ ***r*(*f*, 3)**. We added the predicate *r*(*f*, 3) because the third parenthesis in the part-of-speech string refers to an ontologically relevant element (i.e. the candidate *range* of the *is-composed-of* property).

The second problem is that words that are candidate ranges can be ambiguous, and they often are, especially if they do not belong to the domain glossary *G*. Considering the previous example of the property *depicts*, the word *fruit* is not a term of the AAT glossary, and it has 3 senses in WordNet (the fruit of a plant, the consequence of some action, an amount of product). The property *depicts*, as defined in the CIDOC, simply requires that the range be of type *Entity* (E1). Therefore, all the three senses of *fruit* in WordNet satisfy this constraint. Whenever the range constraints in a relation checker do not allow a full disambiguation, we apply the SSI algorithm (Navigli and Velardi, 2005), a semantic disambiguation algorithm based on structural pattern recognition, available on-line[8]. The algorithm is applied to the words belonging to the segment fragment *f* and is based on the detection of relevant semantic interconnection patterns between the appropriate senses. These patterns are extracted from a lexical knowledge base that merges WordNet with other resources, like word collocations, on-line dictionaries, etc.

For example, in the fragment "depictions of fruit, flowers, and other objects" the following properties are created for the concept *still_ life#1*:

$$R_{depicts}(still\_\ life\#1, fruit\#1)$$
$$R_{depicts}\ (still\_\ life\#1, flower\#2)$$
$$R_{depicts}\ (still\_\ life\#1,\ object\#1)$$

Some of the semantic patterns supporting this sense selection are shown in Figure 2.

A further possibility is that the range of a relation *R* is a concept *instance*. We create concept instances if the word *w* extracted from the fragment *f* is a named entity. For example, the definition of *Venetian lace* is annotated as "Refers to needle lace created **<current-or-former-location>** in Venice**</current-or-former-location>** […]".

As a result, the following triple is produced:

$$R_{has\text{-}current\text{-}or\text{-}former\text{-}location}(Venetian\_lace\#1,\ Venice{:}city\#1)$$

where *Venetian_ lace#1* is the concept label generated for the term *Venetian lace* in the AAT and *Venice* is an instance of the concept *city#1* (*city, metropolis, urban center*) in WordNet.

[8] SSI is an on-line knowledge-based WSD algorithm accessible from http://lcl.di.uniroma1.it/ssi. The on-line version also outputs the detected semantic connections (as those in Figure 2).

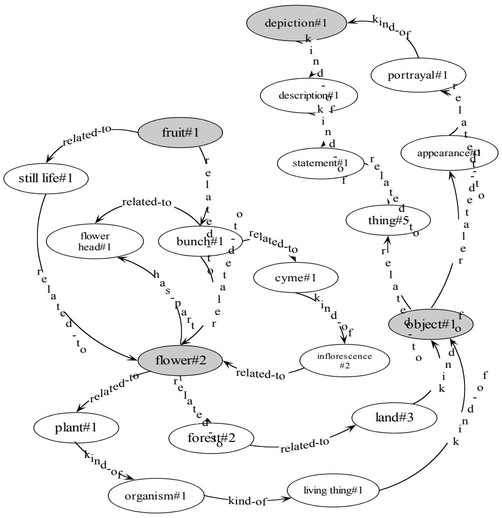


Figure 2. Semantic Interconnections selected by the SSI algorithm when given the word list: "depiction, fruit, flower, object".

4 Evaluation

Since the CIDOC-CRM model formalizes a large number of fine-grained properties (precisely, 141), we selected a subset of properties for our experiments (reported in Table 2). We wrote a relation checker for each property in the Table. By applying the checkers in cascade to a gloss G, a set of annotations is produced. The following is an example of an annotated gloss for the term "vedute":

Refers to detailed, largely factual topographical views, especially **<has-time-span>**18th-century**</has-time-span>** Italian paintings, drawings, or prints of cities. The first vedute probably were **<carried-out-by>**painted by northern European artists**</carried-out-by>** who worked **<has former-or-current-location>**in Italy</**has former-or-current-location**><**has-time-span**>in the 16th century</**has-time-span**>. The term refers more generally to any painting, drawing or print **<depicts>**representing a landscape or town view**</depicts>** that is largely topographical in conception.

Figure 3 shows a more comprehensive graph representation of the outcome for the concepts *vedute*#1 and *maestà*#1 (see the gloss in Section 2.2).

To evaluate the methodology described in Section 3 we considered 814 glosses from the *Visual Works* sub-tree of the AAT thesaurus, containing a total of 27,925 words. The authors wrote the relation checkers by tuning them on a subset of 122 glosses, and tested their generality on the remaining 692. The test set was manually tagged with the subset of the CIDOC-CRM properties shown in Table 2 by two annotators with adjudication (requiring a careful comparison of the two sets of annotations).

We performed two experiments: in the first, we evaluated the *gloss annotation task*, in the second the *property instance extraction task*, i.e. the ability to identify the appropriate domain and range of a property instance. In the case of the gloss annotation task, for evaluating each piece of information we adopted the measures of "*labeled" precision* and *recall*. These measures are commonly used to evaluate parse trees obtained by a parser (Charniak, 1997) and allow the rewarding of good partial results. Given a property *R*, labeled *precision* is the number of *words* annotated correctly with *R* over the number of words annotated automatically with *R*, while labeled *recall* is the number of words annotated correctly with *R* over the total number of words manually annotated with *R*.

Table 3 shows the results obtained by applying the checkers to tag the test set (containing a total number of 1,328 distinct annotations and 5,965 annotated words). Note that here we are evaluating the ability of the system to assign the correct tag to *every word* in a gloss fragment *f*, according to the appropriate relation checker. We choose to evaluate the tag assigned to single words rather than to a whole phrase, because each misalignment would count as a mistake even if the most part of a phrase was tagged correctly by the automatic annotator.

The second experiment consisted in the evaluation of the property instances extracted. Starting from 1,328 manually annotated fragments of 692 glosses, the checkers extracted an overall number of 1,101 property instances. We randomly selected a subset of 160 glosses for evaluation, from which we manually extracted 344 property instances.

Two aspects of the property instance extraction task had to be assessed:

- the extraction of the appropriate *range words* in a gloss, for a given property instance
- the precision and recall in the extraction of the appropriate *concepts* for both *domain* and *range* of the property instance.

An overall number of 233 property instances were automatically collected by the checkers, out of which 203 were correct with respect to the first assessment (87.12% precision (203/233), 59.01% recall (203/344)).

In the second evaluation, for each property instance $R(C_t, C_w)$ we assessed the semantic correctness of both the concepts C_t and C_w. The appropriateness of the concept C_t chosen for the domain must be evaluated, since, even if a term *t* satisfies the semantic constraints of the domain for a property R, still it can be the case that a fragment *f* in *G* does not refer to *t*, like in the following example:

pastels (visual works) -- *Works of art*, typically on a paper or vellum support, to which designs are applied using crayons **made of ground pigment** held together with a binder, typically oil or water and gum.

Code	Name	Domain	Range	Example
P26	moved to	Move	Place	P26(installation of public sculpture, public place)
P27	moved from	Move	Place	P27(removal of cornice pictures, wall)
P53	has former/current location	Physical Stuff	Place	P53(fancy pictures, London)
P55	has current location	Physical Object	Place	P55(macrame, Genoa)
P46	is composed of (is part of)	Physical Stuff	Physical Stuff	P46(lace, knot)
P62	depicts	Physical Man-Made Stuff	Entity	P62(still life, fruit)
P4	has time span	Temporal Entity	Time Span	P4(pattern drawings, Renaissance)
P14	carried out by (performed)	Activity	Actor	P14(blotted line drawings, Andy Warhol)
P92	brought into existence by	Persistent Item	Beginning of Existence	P92(aquatints, aquatint process)
P45	consists of (incorporated in)	Physical Stuff	Material	P45(sculpture, stone)

Table 2: A subset of the relations from the CIDOC-CRM model.

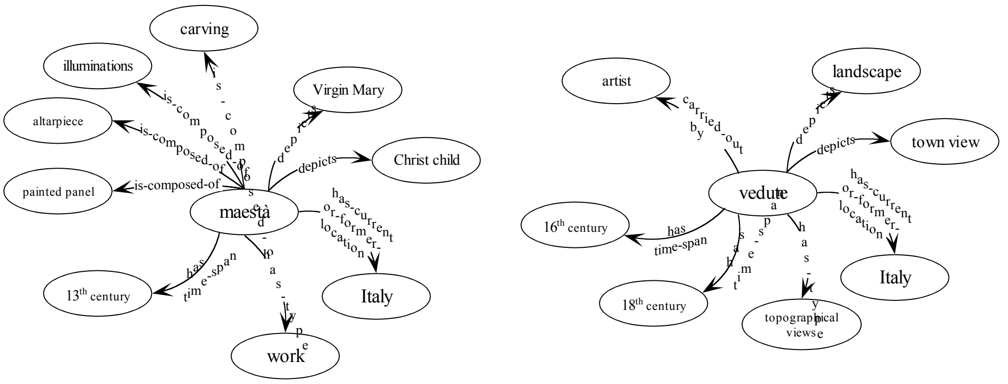


Figure 3. Extracted conceptualisation (in graphical form) of the terms *maestà#1* and *vedute#1* (sense numbers are omitted for clarity).

In this example, *ground pigment* refers to *crayons* (not to *pastels*).

The evaluation of the semantic correctness of the domain and range of the property instances extracted led to the final figures of 81.11% (189/233) precision and 54.94% (189/344) recall, due to 9 errors in the choice of C_t as a domain for an instance $R(C_t, C_w)$ and 5 errors in the semantic disambiguation of range words *w* not appearing in AAT, but encoded in WordNet (as described in the last part of Section 3). A final experiment was performed to evaluate the generality of the approach presented in this paper.

As already remarked, the same procedure used for annotating the glosses of a thesaurus can be used to annotate web documents. Our objective in this third experiment was to:

- Evaluate the ability of the system to annotate fragments of web documents with CIDOC relations
- Evaluate the domain dependency of the relation checkers, by letting the system annotate documents not in the cultural heritage domain.

We then selected 5 documents at random from an historical archive and an artist's biographies archive[9] including about 6,000 words in total, about 5,000 of which in the historical domain. We then ran the automatic annotation procedure on these documents and we evaluated the result, using the same criteria as in Table 3.

Property	Precision	Recall
P26 – moved to	84.95% (79/93)	64.23% (79/123)
P27 – moved from	81.25% (39/48)	78.00% (39/50)
P53 - has former or current location	78.09% (916/1173)	67.80% (916/1351)
P55 – has current location	100.00% (8/8)	100.00% (8/8)
P46 –composed of	87.49% (944/1079)	70.76% (944/1334)
P62 – depicts	94.15% (370/393)	65.26% (370/567)
P4 – has time span	91.93% (547/595)	76.40% (547/716)
P14 - carried out by	91.71% (343/374)	71.91% (343/477)
P92 – brought into existence	89.54% (471/526)	62.72% (471/751)
P45 – consists of	74.67% (398/533)	57.60% (398/691)
Avg. performance	**85.34% (4115/4822)**	**67.81% (4115/6068)**

Table 3: Precision and Recall of the gloss annotation task.

Table 4 presents the results of the experiment. Only 5 out of 10 properties had at least one

[9] http://historicaltextarchive.com and http://www.artnet.com/library

instance in the analysed documents. It is remarkable that, especially for the less domain-dependent properties, the precision and recall of the algorithm is still high, thus showing the generality of the method. Notice that the historical documents influenced the result much more than the artist biographies, because of their dimension.

In Table 4 the recall of P14 (*carried out by*) is omitted. This is motivated by the fact that this property, in a generic domain, corresponds to the *agent* relation ("an active animate entity that voluntarily initiates an action"[10]), while in the cultural heritage domain it has a more narrow interpretation (an example of this relation in the CIDOC handbook is: "the painting of the Sistine Chapel (E7) was *carried out by* Michelangelo Buonarroti (E21) *in the role of* master craftsman (E55)"). However, the domain and range restrictions for P14 correspond to an agent relation, therefore, in a generic domain, one should annotate as "carried out by" almost any verb phrase with the subject (including pronouns and anaphoric references) in the class Human.

Property	**Precision**		**Recall**	
P53 – has former or current location	79.84%	(198/248)	77.95%	(198/254)
P46 – composed of	83.58%	(112/134)	96.55%	(112/116)
P4 – has time span	78.32%	(112/143)	50.68%	(112/221)
P14 – carried out by	60.61%	(40/66)	-	-
P45 – consists of	85.71%	(6/7)	37.50%	(6/16)
Avg. performance	**78.26%**	**(468/598)**	**77.10%**	**(468/607)**

Table 4: Precision and Recall of a web document annotation task.

5 Related work

This paper presented a method to automatically annotate the glosses of a thesaurus, the AAT, with the properties (conceptual relations) of a core ontology, the CIDOC-CRM. Several methods for ontology population and semantic annotation described in literature (e.g. (Thelen and Riloff, 2002; Califf and Mooney, 2004; Cimiano et al. 2005; Valarakos et al. 2004)) use regular expressions to identify named entities, i.e. concept *instances*. Other methods extract hypernym[11] relations using syntactic and lexical patterns (Snow et al. 2005; Morin and Jaquemin 2004) or supervised clustering techniques (Kashyap et al. 2003).

In our work, we automatically learn *formal concepts*, not simply instances or taxonomies (e.g. the graphs of Figure 3) compliant with the semantics of a well-established core ontology, the CIDOC. The method is unsupervised, in the sense that it does not need manual annotation of a significant fragment of text. However, it relies on a set of manually written regular expressions, based on lexical, part-of-speech, and semantic constraints. The structure of regular expressions is rather more complex than in similar works using regular expressions, especially for the use of automatically verified semantic constraints. This complexity is indeed necessary to identify non-trivial relations in an unconstrained text and without training. The issue is however how much this method *generalizes* to other domains:

- A first problem is the availability of lexical and semantic resources used by the algorithm. The most critical requirement of the method is the availability of sound domain *core ontologies*, which hopefully will be produced by other web communities stimulated by the recent success of CIDOC CRM. On the other side, *in absence of an agreed conceptual reference model, no large scale annotation is possible at all.* As for the other resources used by our algorithm, glossaries, thesaura and gazetteers are widely available in "mature" domains. If not, we developed a methodology, described in (Navigli and Velardi, 2005b), to automatically create a glossary in novel domains (e.g. enterprise interoperability), extracting definition sentences from domain-relevant documents and authoritative web sites.
- The second problem is about the generality of regular expressions. Clearly, the relation checkers that we defined are tuned on the CIDOC properties. This however is consistent with our target: in specific domains users are interested to identify specific relations, not general purpose. Certain relevant application domains –like cultural heritage, e-commerce, or tourism- are those that dictate specifications for real-world applications of NLP techniques. However, several CIDOC properties are rather general (especially locative and

[10] http://www.jfsowa.com/ontology/thematic.htm

[11] In AAT the hypernym relation is already available, since AAT is a thesaurus, not a glossary. However we developed regular expressions also for hypernym extraction from definitions. For sake of space this is not discussed in this paper, however the remarkable result (wrt analogous evaluations in literature) is that in 34% of the cases the automatically extracted hypernym is the same as in AAT, and in 26% of the cases, either the extracted hypernym is more general than the one defined in AAT, or the contrary, wrt the AAT hierarchy.

temporal relations) therefore some relation checkers easily apply to other domains, as demonstrated by the experiment on automatic annotation of historical archives in Table 4. Furthermore, the method used to verify semantic constraints is fully general, since it is based on WordNet and a general-purpose, untrained semantic disambiguation algorithm, SSI.

- Finally, the authors believe with some degree of convincement that automatic pattern-learning methods often require non-trivial human effort just like manual methods (because of the need of annotated data, careful parameter setting, etc.), and furthermore they are unable to combine in a non-trivial way different types of features (e.g. lexical, syntactic, semantic). To make an example, a recent work on learning hypernymy patterns (Morin and Jacquemin, 2004) provides the full list of learned patterns. The complexity of these patterns is certainly lower than the regular expression structures used in this work, and many of them are rather intuitive.

In the literature the tasks on which automatic methods have been tested are rather constrained, and do not convincingly demonstrate the superiority of automatic with respect to manually defined patterns. For example, in Senseval-3 (automated labeling of semantic roles[12]), participating systems are requested to identify semantic roles in a sentence fragment for which the "frame semantics" is given, therefore the possible semantic relations to be identified are quite limited.

However, we believe that our method can be automated to some degree (for example, machine learning methods can be used to bootstrap the syntactic patterns, and to learn semantic constraints), a research line we are currently exploring.

References

M. E. Califf and R.J. Mooney, "Bottom-up relational learning of pattern matching rules for information extraction" Machine Learning research, 4 (2)177-210, 2004.

E. Charniak, "Statistical Techniques for Natural Language Parsing", AI Magazine 18(4), 33-44, 1997.

P. Cimiano, G. Ladwig and S. Staab, "Gimme the context: context-driven automatic semantic annotation with C-PANKOW" In: Proceedings of the 14th International WWW Conference, WWW 2005, Chiba, Japan, May, 2005. ACM Press.

M. Doerr, "The CIDOC Conceptual Reference Module: An Ontological Approach to Semantic Interoperability of Metadata". AI Magazine, Volume 24, Number 3, Fall 2003.

M. S. Fox, M. Barbuceanu, M. Gruninger, and J. Lin, "An Organisation Ontology for Enterprise Modeling", In Simulating Organizations: Computational Models of Institutions and Groups, M. Prietula, K. Carley & L. Gasser (Eds), Menlo Park CA: AAAI/MIT Press, pp. 131-152. 1997

J.E. F. Friedl "Mastering Regular Expressions" O'Reilly eds., ISBN: 1-56592-257-3, First edition January 1997.

V. Kashyap, C. Ramakrishnan, T. Rindflesch. "Toward (Semi)-Automatic Generation of Bio-medical Ontologies", Proceedings of American Medical Informatics Association, 2003

G. A. Miller, ``WordNet: a lexical database for English." In: Communications of the ACM 38 (11), November 1995, pp. 39 - 41.

E. Morin and C. Jacquemin "Automatic acquisition and expansion of hypernym links" Computer and the Humanities, 38: 363-396, 2004

R. Navigli, P. Velardi. Learning Domain Ontologies from Document Warehouses and Dedicated Websites, *Computational Linguistics* (30-2), MIT Press, June, 2004.

R. Navigli and P. Velardi, "Structural Semantic Interconnections: a knowledge-based approach to word sense disambiguation", Special Issue-Syntactic and Structural Pattern Recognition, IEEE TPAMI, Volume: 27, Issue: 7, 2005.

R. Navigli, P. Velardi. Automatic Acquisition of a Thesaurus of Interoperability Terms, Proc. of *16th IFAC World Congress*, Praha, Czech Republic, July 4-8th, 2005b.

R. Snow, D. Jurafsky, A. Y. Ng, "Learning syntactic patters for automatic hypernym discovery", NIPS 17, 2005.

M. Thelen, E. Riloff, "A Bootstrapping Method for Learning Semantic Lexicons using Extraction Pattern Contexts", Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2002.

M. Uschold, M. King, S. Moralee and Y. Zorgios, "The Enterprise Ontology", The Knowledge Engineering Review , Vol. 13, Special Issue on Putting Ontologies to Use (eds. Uschold. M. and Tate. A.), 1998

Valarakos, G. Paliouras, V. Karkaletsis, G. Vouros, "Enhancing Ontological Knowledge through Ontology Population and Enrichment" in Proceedings of the 14th EKAW conf., LNAI, Vol. 3257, pp. 144-156, Springer Verlag, 2004.

[12] http://www.clres.com/SensSemRoles.html

Multilingual Ontology Acquisition from Multiple MRDs

Eric Nichols♭, **Francis Bond**♮, **Takaaki Tanaka**♮, **Sanae Fujita**♮, **Dan Flickinger** ♯

♭ Nara Inst. of Science and Technology
Grad. School of Information Science
Nara, Japan
eric-n@is.naist.jp

♮ NTT Communication Science Labs
Natural Language Research Group
Keihanna, Japan
{bond,takaaki,sanae}@cslab.kecl.ntt.co.jp

♯ Stanford University
CSLI
Stanford, CA
danf@csli.stanford.edu

Abstract

In this paper, we outline the development of a system that automatically constructs ontologies by extracting knowledge from dictionary definition sentences using Robust Minimal Recursion Semantics (RMRS). Combining deep and shallow parsing resource through the common formalism of RMRS allows us to extract ontological relations in greater quantity and quality than possible with any of the methods independently. Using this method, we construct ontologies from two different Japanese lexicons and one English lexicon. We then link them to existing, handcrafted ontologies, aligning them at the word-sense level. This alignment provides a representative evaluation of the quality of the relations being extracted. We present the results of this ontology construction and discuss how our system was designed to handle multiple lexicons and languages.

1 Introduction

Automatic methods of ontology acquisition have a long history in the field of natural language processing. The information contained in ontologies is important for a number of tasks, for example word sense disambiguation, question answering and machine translation. In this paper, we present the results of experiments conducted in automatic ontological acquisition over two languages, English and Japanese, and from three different machine-readable dictionaries.

Useful semantic relations can be extracted from large corpora using relatively simple patterns (e.g., (Pantel et al., 2004)). While large corpora often contain information not found in lexicons, even a very large corpus may not include all the familiar words of a language, let alone those words occurring in useful patterns (Amano and Kondo, 1999). Therefore it makes sense to also extract data from machine readable dictionaries (MRDs).

There is a great deal of work on the creation of ontologies from machine readable dictionaries (a good summary is (Wilkes et al., 1996)), mainly for English. Recently, there has also been interest in Japanese (Tokunaga et al., 2001; Nichols et al., 2005). Most approaches use either a specialized parser or a set of regular expressions tuned to a particular dictionary, often with hundreds of rules. Agirre et al. (2000) extracted taxonomic relations from a Basque dictionary with high accuracy using Constraint Grammar together with hand-crafted rules. However, such a system is limited to one language, and it has yet to be seen how the rules will scale when deeper semantic relations are extracted. In comparison, as we will demonstrate, our system produces comparable results while the framework is immediately applicable to any language with the resources to produce RMRS. Advances in the state-of-the-art in parsing have made it practical to use deep processing systems that produce rich syntactic and semantic analyses to parse lexicons. This high level of semantic information makes it easy to identify the relations between words that make up an ontology. Such an approach was taken by the MindNet project (Richardson et al., 1998). However, deep parsing systems often suffer from small lexicons and large amounts of parse ambiguity, making it difficult to apply this knowledge broadly.

Our ontology extraction system uses Robust Minimal Recursion Semantics (RMRS), a formalism that provides a high level of detail while, at the same time, allowing for the flexibility of underspecification. RMRS encodes syntactic information in a general enough manner to make processing of and extraction from syntactic phenomena including coordination, relative clause analy-

Proceedings of the 2nd Workshop on Ontology Learning and Population, pages 10–17,
Sydney, July 2006. ©2006 Association for Computational Linguistics

sis and the treatment of argument structure from verbs and verbal nouns. It provides a common format for naming semantic relations, allowing them to be generalized over languages. Because of this, we are able to extend our system to cover new languages that have RMRS resourses available with a minimal amount of effort. The underspecification mechanism in RMRS makes it possible for us to produce input that is compatible with our system from a variety of different parsers. By selecting parsers of various different levels of robustness and informativeness, we avoid the coverage problem that is classically associated with approaches using deep-processing; using heterogeneous parsing resources maximizes the quality and quantity of ontological relations extracted. Currently, our system uses input from parsers from three levels: with morphological analyzers the shallowest, parsers using Head-driven Phrase Structure Grammars (HPSG) the deepest and dependency parsers providing a middle ground.

Our system was initially developed for one Japanese dictionary (Lexeed). The use of the abstract formalism, RMRS, made it easy to extend to a different Japanese lexicon (Iwanami) and even a lexicon in a different language (GCIDE).

Section 2 provides a description of RMRS and the tools used by our system. The ontological acquisition system is presented in Section 3. The results of evaluating our ontologies by comparison with existing resources are given in Section 4. We discuss our findings in Section 5.

2 Resources

2.1 The Lexeed Semantic Database of Japanese

The Lexeed Semantic Database of Japanese is a machine readable dictionary that covers the most familiar open class words in Japanese as measured by a series of psycholinguistic experiments (Kasahara et al., 2004). Lexeed consists of all open class words with a familiarity greater than or equal to five on a scale of one to seven. This gives 28,000 words divided into 46,000 senses and defined with 75,000 definition sentences. All definition sentences and example sentences have been rewritten to use only the 28,000 familiar open class words. The definition and example sentences have been treebanked with the JACY grammar (§ 2.4.2).

2.2 The Iwanami Dictionary of Japanese

The Iwanami Kokugo Jiten (Iwanami) (Nishio et al., 1994) is a concise Japanese dictionary. A machine tractable version was made available by the Real World Computing Project for the SENSEVAL-2 Japanese lexical task (Shirai, 2003). Iwanami has 60,321 headwords and 85,870 word senses. Each sense in the dictionary consists of a sense ID and morphological information (word segmentation, POS tag, base form and reading, all manually post-edited).

2.3 The Gnu Contemporary International Dictionary of English

The GNU Collaborative International Dictionary of English (GCIDE) is a freely available dictionary of English based on Webster's Revised Unabridged Dictionary (published in 1913), and supplemented with entries from WordNet and additional submissions from users. It currently contains over 148,000 definitions. The version used in this research is formatted in XML and is available for download from `www.ibiblio.org/webster/`.

We arranged the headwords by frequency and segmented their definition sentences into sub-sentences by tokenizing on semicolons (;). This produced a total of 397,460 pairs of headwords and sub-sentences, for an average of slightly less than four sub-sentences per definition sentence. For corpus data, we selected the first 100,000 definition sub-sentences of the headwords with the highest frequency. This subset of definition sentences contains 12,440 headwords with 36,313 senses, covering approximately 25% of the definition sentences in the GCIDE. The GCIDE has the most polysemy of the lexicons used in this research. It averages over 3 senses per word defined in comparison to Lexeed and Iwanami which both have less than 2.

2.4 Parsing Resources

We used Robust Minimal Recursion Semantics (RMRS) designed as part of the Deep Thought project (Callmeier et al., 2004) as the formalism for our ontological relation extraction engine. We used deep-processing tools from the Deep Linguistic Processing with HPSG Initiative (DELPH-IN: `http://www.delph-in.net/`) as well as medium- and shallow-processing tools for Japanese processing (the morphological analyzer

ChaSen and the dependency parser CaboCha) from the Matsumoto Laboratory.

2.4.1 Robust Minimal Recursion Semantics

Robust Minimal Recursion Semantics is a form of flat semantics which is designed to allow deep and shallow processing to use a compatible semantic representation, with fine-grained atomic components of semantic content so shallow methods can contribute just what they know, yet with enough expressive power for rich semantic content including generalized quantifiers (Frank, 2004). The architecture of the representation is based on Minimal Recursion Semantics (Copestake et al., 2005), including a bag of labeled elementary predicates (EPs) and their arguments, a list of scoping constraints which enable scope underspecification, and a handle that provides a hook into the representation.

The representation can be underspecified in three ways: relationships can be omitted (such as quantifiers, messages, conjunctions and so on); predicate-argument relations can be omitted; and predicate names can be simplified. Predicate names are defined in such a way as to be as compatible (predictable) as possible among different analysis engines, using a lemma_pos_subsense naming convention, where the subsense is optional and the part-of-speech (pos) for coarse-grained sense distinctions is drawn from a small set of general types (**n**oun, **v**erb, **s**ahen (verbal noun), . . .). The predicate `unten_s` (運転 *unten* "drive"), for example, is less specific than `unten_s_2` and thus subsumes it. In order to simplify the combination of different analyses, the EPs are indexed to the corresponding character positions in the original input sentence.

Examples of deep and shallow results for the same sentence 自動車を運転する人 *jidōsha wo unten suru hito* "a person who drives a car (lit: car-ACC drive do person)" are given in Figures 1 and 2 (omitting the indexing). Real predicates are prefixed by an under-bar (_). The deep parse gives information about the scope, message types and argument structure, while the shallow parse gives little more than a list of real and grammatical predicates with a hook.

2.4.2 Deep Parsers (JACY, ERG and PET)

For both Japanese and English, we used the PET System for the high-efficiency processing of typed feature structures (Callmeier, 2000). For Japanese, we used JACY (Siegel, 2000), for English we used the English Resource Grammar (ERG: Flickinger 2000).[1]

JACY The JACY grammar is an HPSG-based grammar of Japanese which originates from work done in the Verbmobil project (Siegel, 2000) on machine translation of spoken dialogues in the domain of travel planning. It has since been extended to accommodate written Japanese and new domains (such as electronic commerce customer email and machine readable dictionaries).

The grammar implementation is based on a system of types. There are around 900 lexical types that define the syntactic, semantic and pragmatic properties of the Japanese words, and 188 types that define the properties of phrases and lexical rules. The grammar includes 50 lexical rules for inflectional and derivational morphology and 47 phrase structure rules. The lexicon contains around 36,000 lexemes.

The English Resource Grammar (ERG) The English Resource Grammar (ERG: (Flickinger, 2000)) is a broad-coverage, linguistically precise grammar of English, developed within the Head-driven Phrase Structure Grammar (HPSG) framework, and designed for both parsing and generation. It was also originally launched within the Verbmobil (Wahlster, 2000) spoken language machine translation project for the particular domains of meeting scheduling and travel planning. The ERG has since been substantially extended in both grammatical and lexical coverage, reaching 80-90% coverage of sizeable corpora in two additional domains: electronic commerce customer email and tourism brochures.

The grammar includes a hand-built lexicon of 23,000 lemmas instantiating 850 lexical types, a highly schematic set of 150 grammar rules, and a set of 40 lexical rules, all organized in a rich multiple inheritance hierarchy of some 3000 typed feature structures. Like other DELPH-IN grammars, the ERG can be processed by several parsers and generators, including the LKB (Copestake, 2002) and PET (Callmeier, 2000). Each successful ERG analysis of a sentence or fragment includes a fine-grained semantic representation in MRS.

For the task of parsing the dictionary definitions in GCIDE (the GNU Collaborative Interna-

[1] Both grammars, the LKB and PET are available at <http://www.delph-in.net/>.

$$\begin{bmatrix}
\textsc{text} & \text{自動車 を 運転 する 人} \\
\textsc{top} & h1 \\
\textsc{rels} & \left\{ \begin{array}{l}
\begin{bmatrix} \textbf{proposition\_m\_rel} & \\ \textsc{lbl} & h1 \\ \textsc{arg0} & e2_{\text{tense=present}} \\ \textsc{marg} & h3 \end{bmatrix}
\begin{bmatrix} \textbf{\_jidousha\_n\_rel} & \\ \textsc{lbl} & h4 \\ \textsc{arg0} & x5 \end{bmatrix}
\begin{bmatrix} \textbf{udef\_rel} & \\ \textsc{lbl} & h6 \\ \textsc{arg0} & x5 \\ \textsc{rstr} & h7 \\ \textsc{body} & h8 \end{bmatrix}
\begin{bmatrix} \textbf{\_unten\_s\_rel} & \\ \textsc{lbl} & h9 \\ \textsc{arg0} & e11_{\text{tense=present}} \\ \textsc{arg1} & x10 \\ \textsc{arg2} & x5 \end{bmatrix} \\
\begin{bmatrix} \textbf{\_hito\_n\_rel} & \\ \textsc{lbl} & h12 \\ \textsc{arg0} & x10 \end{bmatrix}
\begin{bmatrix} \textbf{udef\_rel} & \\ \textsc{lbl} & h13 \\ \textsc{arg0} & x10 \\ \textsc{rstr} & h14 \\ \textsc{body} & h15 \end{bmatrix}
\begin{bmatrix} \textbf{proposition\_m\_rel} & \\ \textsc{lbl} & h10001 \\ \textsc{arg0} & e11_{\text{tense=present}} \\ \textsc{marg} & h16 \end{bmatrix}
\begin{bmatrix} \textbf{unknown\_rel} & \\ \textsc{lbl} & h17 \\ \textsc{arg0} & e2_{\text{tense=present}} \\ \textsc{arg} & x10 \end{bmatrix}
\end{array} \right\} \\
\textsc{hcons} & \{h3 \text{ qeq } h17, h7 \text{ qeq } h4, h14 \text{ qeq } h12, h16 \text{ qeq } h9\} \\
\textsc{ing} & \{h12 \text{ ing } h10001\}
\end{bmatrix}$$

Figure 1: RMRS for the Sense 2 of *doraiba-* “driver” (Cabocha/JACY)

$$\begin{bmatrix}
\textsc{text} & \text{自動車 を 運転 する 人} \\
\textsc{top} & h9 \\
\textsc{rels} & \left\{
\begin{bmatrix} \textbf{jidousha\_n\_rel} & \\ \textsc{lbl} & h1 \\ \textsc{arg0} & x2 \end{bmatrix}
\begin{bmatrix} \textbf{o\_p\_rel} & \\ \textsc{lbl} & h3 \\ \textsc{arg0} & u4 \end{bmatrix}
\begin{bmatrix} \textbf{unten\_s\_rel} & \\ \textsc{lbl} & h5 \\ \textsc{arg0} & e6 \end{bmatrix}
\begin{bmatrix} \textbf{suru\_v\_rel} & \\ \textsc{lbl} & h7 \\ \textsc{arg0} & x8 \end{bmatrix}
\begin{bmatrix} \textbf{hito\_n\_rel} & \\ \textsc{lbl} & h9 \\ \textsc{arg0} & x10 \end{bmatrix}
\right\}
\end{bmatrix}$$

Figure 2: RMRS for the Sense 2 of *doraiba-* “driver” (ChaSen)

tional Dictionary of English; see below), the ERG was minimally extended to include two additional fragment rules, for gap-containing VPs and PPs (idiosyncratic to this domain), and additional lexical entries were manually added for all missing words in the alphabetically first 10,000 definition sentences.

These first 10,000 sentences were parsed and then manually tree-banked to provide the training material for constructing the stochastic model used for best-only parsing of the rest of the definition sentences. Using POS-based unknown-word guessing for missing lexical entries, MRSes were obtained for about 75% of the first 100,000 definition sentences.

2.4.3 Medium Parser (CaboCha-RMRS)

For Japanese, we produce RMRS from the dependency parser Cabocha (Kudo and Matsumoto, 2002). The method is similar to that of Spreyer and Frank (2005), who produce RMRS from detailed German dependencies. CaboCha provides fairly minimal dependencies: there are three links (dependent, parallel, apposition) and they link base phrases (Japanese *bunsetsu*), marked with the syntactic and semantic head. The CaboCha-RMRS parser uses this information, along with heuristics based on the parts-of-speech, to produce underspecified RMRSs. CaboCha-RMRS is capable of making use of HPSG resources, including verbal case frames, to further enrich its output. This allows it to produce RMRS that approaches the granularity of the analyses given by HPSG parsers. Indeed, CaboCha-RMRS and JACY give identical parses for the example sentence in Figure 1. One of our motivations in including a medium parser in our system is to extract more relations that require special processing; the flexibility of CaboCha-RMRS and the RMRS formalism make this possible.

2.4.4 Shallow Parser (ChaSen-RMRS)

The part-of-speech tagger, ChaSen (Matsumoto et al., 2000) was used for shallow processing of Japanese. Predicate names were produced by transliterating the pronunciation field and mapping the part-of-speech codes to the RMRS super types. The part-of-speech codes were also used to judge whether predicates were real or grammatical. Since Japanese is a head-final language, the hook value was set to be the handle of the right-most real predicate. This is easy to do for Japanese, but difficult for English.

3 Ontology Construction

We adopt the ontological relation extraction algorithm used by Nichols et al. (2005). Its goal is to identify the semantic head(s) of a dictionary definition sentence – the relation(s) that best summarize it. The algorithm does this by traversing the RMRS structure of a given definition sentence starting at the HOOK (the highest-scoping semantic relationship) and following its argument structure. When the algorithm can proceed no further, it returns the a tuple consisting of the definition word and the word identified by the semantic relation where the algorithm halted. Our extended algorithm has the following characteristics: sentences with only one content-bearing relation are assumed to identify a synonym; special relation processing (§ 3.1) is used to gather meta-information and identify ontological relations; processing of coordination allows for extraction of multiple ontological relations; filtering by part-of-speech screens out unlikely relations (§ 3.2).

3.1 Special Relations

Occasionally, relations which provide ontological meta-information, such as the specification of domain or temporal expressions, or which help identify the type of ontological relation present are encountered. Nichols et al. (2005) identified these as **special relations**. We use a small number of rules to determine where the semantic head is and what ontological relation should be extracted. A sample of the special relations are listed in Table 1. This technique follows in a long tradition of special treatment of certain words that have been shown to be particularly relevant to the task of ontology construction or which are semantically content-free. These words or relations have also be referred to as “empty heads”, “function nouns”, or “relators” in the literature (Wilkes et al., 1996). Our approach generalizes the treatment of these special relations to rules that are portable for any RMRS (modulo the language specific predicate names) giving it portability that cannot be found in approaches that use regular expressions or specialized parsers.

Special Predicate (s)		Ontological
Japanese	English	Relation
isshu, hitotsu	form, kind, one	`hypernym`
ryaku(shou)	abbreviation	`abbreviation`
bubun, ichibu	part, peice	`meronym`
meishou	name	`name`
keishou	’polite name for’	`name:honorific`
zokushou	’slang for’	`name:slang`

Table 1: Special predicates and their associated ontological relations

Augmenting the system to work on English definition sentence simply entailed writing rules to handle special relations that occur in English. Our system currently has 26 rules for Japanese and 50 rules for English. These rules provide processing of relations like those found in Table 1, and they also handle processing of coordinate structures, such as noun phrases joined together with conjunctions such as *and*, *or*, and punctuation.

3.2 Filtering by Part-of-Speech

One of the problems encountered in expanding the approach in Nichols et al. (2005) to handle English dictionaries is that many of the definition sentences have a semantic head with a part-of-speech different than that of the definition word. We found that differing parts-of-speech often indicated an undesirable ontological relation. One reason such relations can be extracted is when a sentence with a non-defining role, for example indicating usage, is encountered. Definition sentence for non-content-bearing words such as *of* or *the* also pose problems for extraction.

We avoid these problems by filtering by parts-of-speech twice in the extraction process. First, we select candidate sentences for extraction by verifying that the definition word has a content word POS (i.e. adjective, adverb, noun, or verb). Finally, before we extract any ontological relation, we make sure that the definition word and the semantic head are in compatible POS classes.

While adopting this strategy does reduce the number of total ontological relations that we acquire, it increases their reliability. The addition of a medium parser gives us more RMRS structures to extract from, which helps compensate for any loss in number.

4 Results and Evaluation

We summarize the relationships acquired in Table 2. The columns specify source dictionary and parsing method while the rows show the relation type. These counts represent the total number of relations extracted for each source and method combination. The majority of relations extracted are synonyms and hypernyms; however, some higher-level relations such as meronym and abbreviation are also acquired. It should also be noted that both the medium and deep methods were able to extract a fair number of special relations. In many cases, the medium method even extracted more special relations than the deep method. This is yet another indication of the flexibility of dependency parsing. Altogether, we extracted 105,613 unique relations from Lexeed (for 46,000 senses), 183,927 unique relations from Iwanami (for 85,870 senses), and 65,593 unique relations from GCIDE (for 36,313 senses). As can be expected, a general pattern in our results is that the shallow method extracts the most relations in total followed by the medium method, and finally

Relation	Lexeed			Iwanami			GCIDE
	Shallow	Medium	Deep	Shallow	Medium	Deep	Deep
`hypernym`	47,549	43,006	41,553	113,120	113,433	66,713	40,583
`synonym`	12,692	13,126	9,114	31,682	32,261	18,080	21,643
`abbreviation`		340	429		1,533	739	
`meronym`		235	189		395	202	472
`name`		100	89		271	140	

Table 2: Results of Ontology Extraction

the deep method.

4.1 Verification with Hand-crafted Ontologies

Because we are interested in comparing lexical semantics across languages, we compared the extracted ontology with resources in both the same and different languages.

For Japanese we verified our results by comparing the hypernym links to the manually constructed Japanese ontology Goi-Taikei (**GT**). It is a hierarchy of 2,710 semantic classes, defined for over 264,312 nouns Ikehara et al. (1997). The semantic classes are mostly defined for nouns (and verbal nouns), although there is some information for verbs and adjectives. For English, we compared relations to WordNet 2.0 (Fellbaum, 1998). Comparison for hypernyms is done as follows: look up the semantic class or synset C for both the headword (w_i) and genus term(s) (w_g). If at least one of the index word's classes is subsumed by at least one of the genus' classes, then we consider the relationship confirmed (1).

$$\exists(c_h, c_g) : \{c_h \subset c_g; c_h \in C(w_h); c_g \in C(w_g)\} \quad (1)$$

To test cross-linguistically, we looked up the headwords in a translation lexicon (**ALT-J/E** (Ikehara et al., 1991) and EDICT (Breen, 2004)) and then did the confirmation on the set of translations $c_i \subset C(T(w_i))$. Although looking up the translation adds noise, the additional filter of the relationship triple effectively filters it out again.

The total figures given in Table 3 do not match the totals given in Table 2. These totals represent the number of relations where both the definition word and semantic head were found in at least one of the ontologies being used in this comparison. By comparing these numbers to the totals given in Section 4, we can get an idea of the coverage of the ontologies being used in comparison. Lexeed has a coverage of approx. 55.74% ($\frac{58{,}867}{105{,}613}$), with Iwanami the lowest at 48.20% ($\frac{88{,}662}{183{,}927}$), and GCIDE the highest at 69.85% ($\frac{45{,}814}{65{,}593}$). It is clear that there are a lot of relations in each lexicon that are not covered by the hand-crafted ontologies. This demonstrates that machine-readable dictionaries are still a valuable resource for constructing ontologies.

4.1.1 Lexeed

Our results using JACY achieve a confirmation rate of **66.84%** for nouns only and **60.67%** overall (Table 3). This is an improvement over both Tokunaga et al. (2001), who reported 61.4% for nouns only, and Nichols et al. (2005) who reported 63.31% for nouns and 57.74% overall. We also achieve an impressive 33,333 confirmed relations for a rate of 56.62% overall. It is important to note that our total counts include all unique relations regardless of source, unlike Nichols et al. (2005) who take only the relation from the deepest source whenever multiple relations are extracted. It is interesting to note that shallow processing out performs medium with 22,540 verified relations (59.40%) compared to 21,806 (57.76%). This would seem to suggest that for the simplest task of retrieving hyperynms and synonyms, more information than that is not necessary. However, since medium and deep parsing obtain relations not covered by shallow parsing and can extract special relations, a task that cannot be performed without syntactic information, it is beneficial to use them as well.

Agirre et al. (2000) reported an error rate of 2.8% in a hand-evaluation of the semantic relations they automatically extracted from a machine-readable Basque dictionary. In a similar hand-evaluation of a stratified sampling of relations extracted from Lexeed, we achieved an error rate of 9.2%, demonstrating that our method is also highly accurate (Nichols et al., 2005).

4.2 Iwanami

Iwanami's verification results are similar to Lexeed's (Table 3). There are on average around 3% more verifications and a total of almost 20,000 more verified relations extracted. It is particularly interesting to note that deep processing per-

Confirmed Relations in Lexeed

Method / Relation	hypernym	synonym	Total
Shallow	58.55 % (16585 / 28328)	61.93 % (5955 / 9615)	59.40 % (22540 / 37943)
Medium	55.97 % (15431 / 27570)	62.61 % (6375 / 10182)	57.76 % (21806 / 37752)
Deep	54.78 % (4954 / 9043)	67.76 % (5098 / 7524)	60.67 % (10052 / 16567)
All	55.22 % (23802 / 43102)	60.46 % (9531 / 15765)	56.62 % (33333 / 58867)

Confirmed Relations in Iwanami

Method / Relation	hypernym	synonym	Total
Shallow	61.20 % (35208 / 57533)	63.57 % (11362 / 17872)	61.76 % (46570 / 75405)
Medium	60.69 % (35621 / 58698)	62.86 % (11037 / 17557)	61.19 % (46658 / 76255)
Deep	63.59 % (22936 / 36068)	64.44 % (8395 / 13027)	63.82 % (31331 / 49095)
All	59.36 % (40179 / 67689)	61.66 % (12931 / 20973)	59.90 % (53110 / 88662)

Confirmed Relations in GCIDE

POS / Relation	hypernym	synonym	Total
Adjective	2.88 % (37 / 1283)	16.77 % (705 / 4203)	13.53 % (742 / 5486)
Noun	57.60 % (7518 / 13053)	50.71 % (3522 / 6945)	55.21 % (11040 / 19998)
Verb	24.22 % (3006 / 12411)	21.40 % (1695 / 7919)	23.12 % (4701 / 20330)
Total	39.48 % (10561 / 26747)	31.06 % (5922 / 19067)	35.98 % (16483 / 45814)

Table 3: Confirmed Relations, measured against **GT** and WordNet

forms better here than on Lexeed (63.82% vs 60.67%), even though the grammar was developed and tested on Lexeed. There are two reasons for this: The first is that the process of rewriting Lexeed to use only familiar words actually makes the sentences harder to parse. The second is that the less familiar words in Iwanami have fewer senses, and easier to parse definition sentences. In any case, the results support our claims that our ontological relation extraction system is easily adaptable to new lexicons.

4.3 GCIDE

At first glance, it would seem that GCIDE has the most disappointing of the verification results with overall verification of not even 36% and only 16,483 relations confirmed. However, on closer inspection one can see that noun hypernyms are a respectable 57.60% with over 55% for all nouns. These figures are comparable with the results we are obtaining with the other lexicons. One should also bear in mind that the definitions found in GCIDE can be archaic; after all this dictionary was first published in 1913. This could be one cause of parsing errors for ERG. Despite these obstacles, we feel that GCIDE has a lot of potential for ontological acquisition. A dictionary of its size and coverage will most likely contain relations that may not be represented in other sources. One only has to look at the definition of ドライバー "driver"/*driver* to confirm this; **GT** has two senses ("screwdriver" and "vehicle operator") Lexeed and Iwanami have 3 senses each (adding "golf club"), and WordNet has 5 (including "software driver"), but GCIDE has 6, not including "software driver" but including *spanker* "a kind of sail". It should be beneficial to propagate these different senses across ontologies.

5 Discussion and Future Work

We were able to successfully combine deep processing of various levels of depth in order to extract ontological information from lexical resources. We showed that, by using a well defined semantic representation, the extraction can be generalized so much that it can be used on very different dictionaries from different languages. This is an improvement on the common approach to using more and more detailed regular expressions (e.g. Tokunaga et al. (2001)). Although this provides a quick start, the results are not generally reusable. In comparison, the shallower RMRS engines are immediately useful for a variety of other tasks.

However, because the hook is the only syntactic information returned by the shallow parser, ontological relation extraction is essentially performed by this hook-identifying heuristic. While this is sufficient for a large number of sentences, it is not possible to process special relations with the shallow parser since none of the arguments are linked with the predicates to which they belong. Thus, as Table 2 shows, our shallow parser is only capable of retrieving hypernyms and synonyms. It is important to extract a variety of semantic relations in order to form a useful ontology. This is one of the reasons why we use a combination of parsers of different analytic levels rather than depending on a single resource.

The other innovation of our approach is the cross-lingual evaluation. As a by-product of the evaluation we enhance the existing resources (such as **GT** or WordNet) by linking them, so that information can be shared between them. In this way we can use the cross-lingual links to fill gaps in the monolingual resources. **GT** and WordNet both lack complete cover - over half the relations were confirmed with only one resource. This shows that the machine readable dictionary is a useful source of these relations.

6 Conclusion

In this paper, we presented the results of experiments conducted in automatic ontological acquisition over two languages, English and Japanese, and from three different machine-readable dictionaries. Our system is unique in combining parsers of various levels of analysis to generate its input semantic structures. The system is language agnostic and we give results for both Japanese and English MRDs. Finally, we presented evaluation of the ontologies constructed by comparing them with existing hand-crafted English and Japanese ontologies.

References

Eneko Agirre, Olatz Ansa, Xabier Arregi, Xabier Artola, Arantza Diaz de Ilarraza, Mikel Lersundi, David Martinez, Kepa Sarasola, and Ruben Urizar. 2000. Extraction of semantic relations from a Basque monolingual dictionary using Constraint Grammar. In *EURALEX 2000*.

Shigeaki Amano and Tadahisa Kondo. 1999. *Nihongo-no Goi-Tokusei (Lexical properties of Japanese)*. Sanseido.

J. W. Breen. 2004. JMDict: a Japanese-multilingual dictionary. In *Coling 2004 Workshop on Multilingual Linguistic Resources*, pages 71–78. Geneva.

Ulrich Callmeier. 2000. PET - a platform for experimentation with efficient HPSG processing techniques. *Natural Language Engineering*, 6(1):99–108.

Ulrich Callmeier, Andreas Eisele, Ulrich Schäfer, and Melanie Siegel. 2004. The DeepThought core architecture framework. In *Proceedings of LREC-2004*, volume IV. Lisbon.

Ann Copestake. 2002. *Implementing Typed Feature Structure Grammars*. CSLI Publications.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal Recursion Semantics. An introduction. *Research on Language and Computation*, 3(4):281–332.

Christine Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Dan Flickinger. 2000. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28. (Special Issue on Efficient Processing with HPSG).

Anette Frank. 2004. Constraint-based RMRS construction from shallow grammars. In *20th International Conference on Computational Linguistics: COLING-2004*, pages 1269–1272. Geneva.

Satoru Ikehara, Masahiro Miyazaki, Satoshi Shirai, Akio Yokoo, Hiromi Nakaiwa, Kentaro Ogura, Yoshifumi Ooyama, and Yoshihiko Hayashi. 1997. *Goi-Taikei — A Japanese Lexicon*. Iwanami Shoten, Tokyo. 5 volumes/CDROM.

Satoru Ikehara, Satoshi Shirai, Akio Yokoo, and Hiromi Nakaiwa. 1991. Toward an MT system without pre-editing — effects of new methods in **ALT-J/E** —. In *Third Machine Translation Summit: MT Summit III*, pages 101–106. Washington DC. (`http://xxx.lanl.gov/abs/cmp-lg/9510008`).

Kaname Kasahara, Hiroshi Sato, Francis Bond, Takaaki Tanaka, Sanae Fujita, Tomoko Kanasugi, and Shigeaki Amano. 2004. Construction of a Japanese semantic lexicon: Lexeed. SIG NLC-159, IPSJ, Tokyo. (in Japanese).

Taku Kudo and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops)*, pages 63–69. Taipei.

Yuji Matsumoto, Kitauchi, Yamashita, Hirano, Matsuda, and Asahara. 2000. *Nihongo Keitaiso Kaiseki System: Chasen*. `http://chasen.naist.jp/hiki/ChaSen/`.

Eric Nichols, Francis Bond, and Daniel Flickinger. 2005. Robust ontology acquisition from machine-readable dictionaries. In *Proceedings of the International Joint Conference on Artificial Intelligence IJCAI-2005*, pages 1111–1116. Edinburgh.

Minoru Nishio, Etsutaro Iwabuchi, and Shizuo Mizutani. 1994. *Iwanami Kokugo Jiten Dai Go Han [Iwanami Japanese Dictionary Edition 5]*. Iwanami Shoten, Tokyo. (in Japanese).

Patrick Pantel, Deepak Ravichandran, and Eduard Hovy. 2004. Towards terascale knowledge acquisition. In *20th International Conference on Computational Linguistics: COLING-2004*, pages 771–777. Geneva.

Stephen D. Richardson, William B. Dolan, and Lucy Vanderwende. 1998. MindNet: acquiring and structuring semantic information from text. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics: COLING/ACL-98*, pages 1098–1102. Montreal.

Kiyoaki Shirai. 2003. SENSEVAL-2 Japanese dictionary task. *Journal of Natural Language Processing*, 10(3):3–24. (in Japanese).

Melanie Siegel. 2000. HPSG analysis of Japanese. In Wahlster (2000), pages 265 – 280.

Kathrin Spreyer and Anette Frank. 2005. The TIGER RMRS 700 bank: RMRS construction from dependencies. In *Proceedings of the 6th International Workshop on Linguistically Interpreted Corpora (LINC 2005)*, pages 1–10. Jeju Island, Korea.

Takenobu Tokunaga, Yasuhiro Syotu, Hozumi Tanaka, and Kiyoaki Shirai. 2001. Integration of heterogeneous language resources: A monolingual dictionary and a thesaurus. In *Proceedings of the 6th Natural Language Processing Pacific Rim Symposium, NLPRS2001*, pages 135–142. Tokyo.

Wolfgang Wahlster, editor. 2000. *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer, Berlin, Germany.

Yorick A. Wilkes, Brian M. Slator, and Louise M. Guthrie. 1996. *Electric Words*. MIT Press.

LEILA: Learning to Extract Information by Linguistic Analysis

Fabian M. Suchanek
Max-Planck-Institute
for Computer Science
Saarbrücken/Germany
suchanek@mpii.mpg.de

Georgiana Ifrim
Max-Planck-Institute
for Computer Science
Saarbrücken/Germany
ifrim@mpii.mpg.de

Gerhard Weikum
Max-Planck-Institute
for Computer Science
Saarbrücken/Germany
weikum@mpii.mpg.de

Abstract

One of the challenging tasks in the context of the Semantic Web is to automatically extract instances of binary relations from Web documents – for example all pairs of a person and the corresponding birthdate. In this paper, we present LEILA, a system that can extract instances of arbitrary given binary relations from natural language Web documents – without human interaction. Different from previous approaches, LEILA uses a deep syntactic analysis. This results in consistent improvements over comparable systems (such as e.g. Snowball or TextToOnto).

1 Introduction

1.1 Motivation

Search engines, question answering systems and classification systems alike can greatly profit from formalized world knowledge. Unfortunately, manually compiled collections of world knowledge (such as e.g. WordNet (Fellbaum, 1998)) often suffer from low coverage, high assembling costs and fast aging. In contrast, the World Wide Web provides an enormous source of knowledge, assembled by millions of people, updated constantly and available for free. Since the Web data consists mostly of natural language documents, a first step toward exploiting this data would be to extract instances of given target relations. For example, one might be interested in extracting all pairs of a person and her birthdate (the `birthdate`-relation), pairs of a company and the city of its headquarters (the `headquarters`-relation) or pairs of an entity and the concept it belongs to (the `instanceOf`-relation). The task is, given a set of Web documents and given a target relation, extracting pairs of entities that are in the target relation. In this paper, we propose a novel method for this task, which works on natural language Web documents and does not require human interaction. Different from previous approaches, our approach involves a deep linguistic analysis, which helps it to achieve a superior performance.

1.2 Related Work

There are numerous Information Extraction (IE) approaches, which differ in various features:

- **Arity of the target relation:** Some systems are designed to extract unary relations, i.e. sets of entities (Finn and Kushmerick, 2004; Califf and Mooney, 1997). In this paper we focus on the more general binary relations.
- **Type of the target relation:** Some systems are restricted to learning a single relation, mostly the `instanceOf`-relation (Cimiano and Völker, 2005b; Buitelaar et al., 2004). In this paper, we are interested in extracting arbitrary relations (including `instanceOf`). Other systems are designed to discover new binary relations (Maedche and Staab, 2000). However, in our scenario, the target relation is given in advance.
- **Human interaction:** There are systems that require human intervention during the IE process (Riloff, 1996). Our work aims at a completely automated system.
- **Type of corpora:** There exist systems that can extract information efficiently from formatted data, such as HTML-tables or structured text (Graupmann, 2004; Freitag and Kushmerick, 2000). However, since a large part of the Web consists of natural language text, we consider in this paper only systems that accept also unstructured corpora.
- **Initialization:** As initial input, some systems require a hand-tagged corpus (J. Iria, 2005; Soderland et al., 1995), other systems require text patterns (Yangarber et al., 2000) or templates (Xu and Krieger, 2003) and again others require seed tuples (Agichtein and Gravano, 2000; Ruiz-Casado et al., 2005; Mann and Yarowsky, 2005) or tables of target concepts (Cimiano and Völker, 2005a). Since hand-

Proceedings of the 2nd Workshop on Ontology Learning and Population, pages 18–25,
Sydney, July 2006. ©2006 Association for Computational Linguistics

labeled data and manual text patterns require huge human effort, we consider only systems that use seed pairs or tables of concepts.

Furthermore, there exist systems that use the whole Web as a corpus (Etzioni et al., 2004) or that validate their output by the Web (Cimiano et al., 2005). In order to study different extraction techniques in a controlled environment, however, we restrict ourselves to systems that work on a closed corpus for this paper.

One school of **extraction techniques** concentrates on detecting the boundary of interesting entities in the text, (Califf and Mooney, 1997; Finn and Kushmerick, 2004; Yangarber et al., 2002). This usually goes along with the restriction to unary target relations. Other approaches make use of the context in which an entity appears (Cimiano and Völker, 2005a; Buitelaar and Ramaka, 2005). This school is mostly restricted to the `instanceOf`-relation. The only group that can learn arbitrary binary relations is the group of pattern matching systems (Etzioni et al., 2004; Agichtein and Gravano, 2000; Ravichandran and Hovy, 2002; Brin, 1999; Soderland, 1999; Xu et al., 2002; Ruiz-Casado et al., 2005; Mann and Yarowsky, 2005). Surprisingly, none of these systems uses a deep linguistic analysis of the corpus. Consequently, most of them are extremely volatile to small variations in the patterns. For example, the simple subordinate clause in the following example (taken from (Ravichandran and Hovy, 2002)) can already prevent a surface pattern matcher from discovering a relation between ”London” and the ”river Thames”: ”London, which has one of the busiest airports in the world, lies on the banks of the river Thames.”

1.3 Contribution

This paper presents LEILA (Learning to Extract Information by Linguistic Analysis), a system that can extract instances of an arbitrary given binary relation from natural language Web documents without human intervention. LEILA uses a deep analysis for natural-language sentences as well as other advanced NLP methods like anaphora resolution, and combines them with machine learning techniques for robust and high-yield information extraction. Our experimental studies on a variety of corpora demonstrate that LEILA achieves very good results in terms of precision and recall and outperforms the prior state-of-the-art methods.

1.4 Link Grammars

There exist different approaches for parsing natural language sentences. They range from simple part-of-speech tagging to context-free grammars and more advanced techniques such as Lexical Functional Grammars, Head-Driven Phrase Structure Grammars or stochastic approaches. For our implementation, we chose the Link Grammar Parser (Sleator and Temperley, 1993). It is based on a context-free grammar and hence it is simpler to handle than the advanced parsing techniques. At the same time, it provides a much deeper semantic structure than the standard context-free parsers. Figure 1 shows a simplified example of a linguistic structure produced by the link parser (a *linkage*).

A linkage is a connected planar undirected graph, the nodes of which are the words of the sentence. The edges are called *links*. They are labeled with *connectors*. For example, the connector **subj** in Figure 1 marks the link between the subject and the verb of the sentence. The linkage must fulfill certain linguistic constraints, which are given by a *link grammar*. The link grammar specifies which word may be linked by which connector to preceding and following words. Furthermore, the parser assigns part-of-speech tags, i.e. symbols identifying the grammatical function of a word in a sentence. In the example in Figure 1, the letter ”n” following the word ”composers” indentifies ”composers” as a noun.

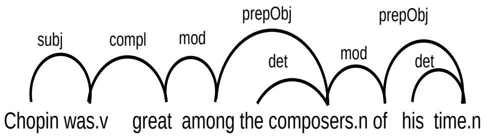


Figure 1: A simple linkage

Figure 2 shows how the Link Parser copes with a more complex example. The relationship between the subject ”London” and the verb ”lies” is not disrupted by the subordinate clause:

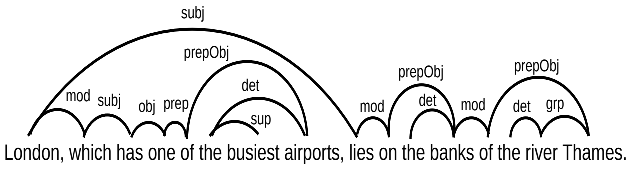

subj
prepObj
det
mod
subj
obj
prep
sup
mod
prepObj
det
mod
prepObj
det
grp
London, which has one of the busiest airports, lies on the banks of the river Thames.

Figure 2: A complex linkage

We say that a linkage *expresses* a relation r, if the underlying sentence implies that a pair of entities is in r. Note that the deep grammatical analysis of the sentence would allow us to define the meaning of the sentence in a theoretically well-founded way (Montague, 1974). For this paper, however, we limit ourselves to an intuitive understanding of the notion of meaning.

We define a *pattern* as a linkage in which two words have been replaced by placeholders. Figure 3 shows a pattern derived from the linkage in Figure 1 by replacing ”Chopin” and ”composers” by the placeholders ”X” and ”Y”.

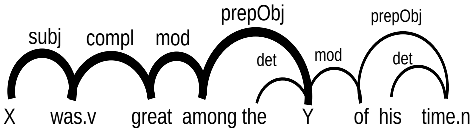


Figure 3: A pattern

We call the (unique) shortest path from one placeholder to the other the *bridge*, marked in bold in the figure. The bridge does not include the placeholders. Two bridges are regarded as equivalent, if they have the same sequence of nodes and edges, although nouns and adjectives are allowed to differ. For example, the bridge in Figure 3 and the bridge in Figure 4 (in bold) are regarded as equivalent, because they are identical except for a substitution of ”great” by ”mediocre”. A pattern *matches* a linkage, if an equivalent bridge occurs in the linkage. For example, the pattern in Figure 3 matches the linkage in Figure 4.

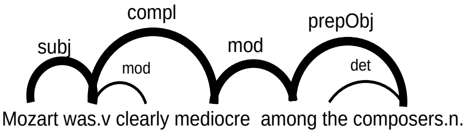


Figure 4: A matching linkage

If a pattern matches a linkage, we say that the pattern *produces* the pair of words that the linkage contains in the position of the placeholders. In Figure 4, the pair ”Mozart” / ”composers” is produced by the pattern in Figure 3.

2 System Description

2.1 Document Pre-Processing

LEILA accepts HTML documents as input. To allow the system to handle date and number expressions, we normalize these constructions by regular expression matching in combination with a set of functions. For example, the expression ”November 23rd to 24th 1998” becomes ”1998-11-23 to 1998-11-24” and the expression ”0.8107 acre-feet” becomes ”1000 cubic-meters”. Then, we split the original HTML-document into two files: The first file contains the proper sentences with the HTML-tags removed. The second file contains the non-grammatical parts, such as lists, expressions using parentheses and other constructions that cannot be handled by the Link Parser. For example, the character sequence ”Chopin (born 1810) was a great composer” is split into the sentence ”Chopin was a great composer” and the non-grammatical information ”Chopin (born 1810)”. The grammatical file is parsed by the Link Parser.

The parsing allows for a restricted named entity recognition, because the parser links noun groups like ”United States of America” by designated connectors. Furthermore, the parsing allows us to do anaphora resolution. We use a conservative approach, which simply replaces a third person pronoun by the subject of the preceding sentence. For our goal, it is essential to normalize nouns to their singular form. This task is non-trivial, because there are numerous words with irregular plural forms and there exist even word forms that can be either the singular form of one word or the plural form of another. By collecting these exceptions systematically from WordNet, we were able to stem most of them correctly with our Plural-to-Singular Stemmer (*PlingStemmer*[1]). For the non-grammatical files, we provide a pseudo-parsing, which links each two adjacent items by an artificial connector. As a result, the uniform output of the preprocessing is a sequence of linkages, which constitutes the input for the core algorithm.

2.2 Core Algorithm

As a definition of the target relation, our algorithm requires a function (given by a Java method) that decides into which of the following categories a pair of words falls:

- The pair can be an **example** for the target relation. For instance, for the `birthdate`-relation, the examples can be given by a list of persons with their birth dates.
- The pair can be a **counterexample**. For the `birthdate`-relation, the counterexamples can be deduced from the examples (e.g. if ”Chopin” / ”1810” is an example, then ”Chopin” / ”2000” must be a counterexample).
- The pair can be a **candidate**. For `birthdate`, the candidates would be all pairs of a proper name and a date that are not an example or a counterexample.
- The pair can be none of the above.

The core algorithm proceeds in three phases:

1. In the *Discovery Phase*, it seeks linkages in which an example pair appears. It replaces the two words by placeholders, thus producing a pattern. These patterns are collected as *positive patterns*. Then, the algorithm runs through the sentences again and finds all linkages that match

[1] available at http://www.mpii.mpg.de/ ∼suchanek

a positive pattern, but produce a counterexample. The corresponding patterns are collected as *negative patterns*[2].

2. In the *Training Phase*, statistical learning is applied to learn the concept of positive patterns. The result of this process is a classifier for patterns.
3. In the *Testing Phase*, the algorithm considers again all sentences in the corpus. For each linkage, it generates all possible patterns by replacing two words by placeholders. If the two words form a candidate and the pattern is classified as positive, the produced pair is proposed as a new element of the target relation (an *output pair*).

In principle, the core algorithm does not depend on a specific grammar or a specific parser. It can work on any type of grammatical structures, as long as some kind of pattern can be defined on them. It is also possible to run the Discovery Phase and the Testing Phase on different corpora.

2.3 Learning Model

The central task of the Discovery Phase is determining patterns that express the target relation. These patterns are generalized in the Training Phase. In the Testing Phase, the patterns are used to produce the output pairs. Since the linguistic meaning of the patterns is not apparent to the system, the Discovery Phase relies on the following hypothesis: Whenever an example pair appears in a sentence, the linkage and the corresponding pattern express the target relation. This hypothesis may fail if a sentence contains an example pair merely by chance, i.e. without expressing the target relation. Analogously, a pattern that does express the target relation may occasionally produce counterexamples. We call these patterns *false samples*. Virtually any learning algorithm can deal with a limited number of false samples.

To show that our approach does not depend on a specific learning algorithm, we implemented two classifiers for LEILA: One is an adaptive k-Nearest-Neighbor-classifier (kNN) and the other one uses a Support Vector Machine (SVM). These classifiers, the feature selection and the statistical model are explained in detail in (Suchanek et al., 2006). Here, we just note that the classifiers yield a real valued label for a test pattern. This value can be interpreted as the confidence of the classification. Thus, it is possible to rank the output pairs of LEILA by their confidence.

3 Experiments

3.1 Setup

We ran LEILA on different corpora with increasing heterogeneity:

- **Wikicomposers:** The set of all Wikipedia articles about composers (872 HTML documents). We use it to see how LEILA performs on a document collection with a strong structural and thematic homogeneity.
- **Wikigeography:** The set of all Wikipedia pages about the geography of countries (313 HTML documents).
- **Wikigeneral:** A set of random Wikipedia articles (78141 HTML documents). We chose it to assess LEILA's performance on structurally homogenous, but thematically random documents.
- **Googlecomposers:** This set contains one document for each baroque, classical, and romantic composer in Wikipedia's list of composers, as delivered by a Google ”I'm feeling lucky” search for the composer's name (492 HTML documents). We use it to see how LEILA performs on a corpus with a high structural heterogeneity. Since the querying was done automatically, the downloaded pages include spurious advertisements as well as pages with no proper sentences at all.

We tested LEILA on different target relations with increasing complexity:

- **birthdate:** This relation holds between a person and his birth date (e.g. ”Chopin” / ”1810”). It is easy to learn, because it is bound to strong surface clues (the first element is always a name, the second is always a date).
- **synonymy:** This relation holds between two names that refer to the same entity (e.g. ”UN”/”United Nations”). The relation is more sophisticated, since there are no surface clues.
- **instanceOf:** This relation is even more sophisticated, because the sentences often express it only implicitly.

We compared LEILA to different **competitors**. We only considered competitors that, like LEILA, extract the information from a corpus without using other Internet sources. We wanted to avoid running the competitors on our own corpora or on our own target relations, because we could not be sure to achieve a fair tuning of the competitors. Hence we ran LEILA on the corpora and the target relations that our competitors have been tested on by their authors. We compare the results of LEILA with the results reported by the authors. Our competitors, together with their respective corpora and relations, are:

[2] Note that different patterns can match the same linkage.

- **TextToOnto[3]:** A state-of-the-art representative for non-deep pattern matching. The system provides a component for the `instanceOf` relation and takes arbitrary HTML documents as input. For completeness, we also consider its successor Text2Onto (Cimiano and Völker, 2005a), although it contains only default methods in its current state of development.
- **Snowball (Agichtein and Gravano, 2000):** A recent representative of the slot-extraction paradigm. In the original paper, Snowball has been tested on the `headquarters` relation. This relation holds between a company and the city of its headquarters. Snowball was trained on a collection of some thousand documents and then applied to a test collection. For copyright reasons, we only had access to the test collection (150 text documents).
- (Cimiano and Völker, 2005b) present a new system that uses context to assign a concept to an entity. We will refer to this system as the **CV-system**. The approach is restricted to the `instanceOf`-relation, but it can classify instances even if the corpus does not contain explicit definitions. In the original paper, the system was tested on a collection of 1880 files from the Lonely Planet Internet site[4].

For the **evaluation**, the output pairs of the system have to be compared to a table of ideal pairs. One option would be to take the ideal pairs from a pre-compiled data base. The problem is that these ideal pairs may differ from the facts expressed in the documents. Furthermore, these ideal pairs do not allow to measure how much of the document content the system actually extracted. This is why we chose to extract the ideal pairs manually from the documents. In our methodology, the ideal pairs comprise all pairs that a human would understand to be elements of the target relation. This involves full anaphora resolution, the solving of reference ambiguities, and the choice of truly defining concepts. For example, we accept Chopin as instance of `composer` but not as instance of `member`, even if the text says that he was a member of some club. Of course, we expect neither the competitors nor LEILA to achieve the results in the ideal table. However, this methodology is the only fair way of manual extraction, as it is guaranteed to be system-independent. If O denotes the multi-set of the output pairs and I denotes the multi-set of the ideal pairs, then precision, recall, and their harmonic mean $F1$ can be computed as

$$recall = \frac{|O \cap I|}{|I|} \qquad precision = \frac{|O \cap I|}{|O|}$$

$$F1 = \frac{2 \times recall \times precision}{recall + precision} \quad .$$

To ensure a fair comparison of LEILA to Snowball, we use the same evaluation as employed in the original Snowball paper (Agichtein and Gravano, 2000), the *Ideal Metric*. The Ideal Metric assumes the target relation to be right-unique (i.e. a many-to-one relation). Hence the set of ideal pairs is right-unique. The set of output pairs can be made right-unique by selecting the pair with the highest confidence for each first component. Duplicates are removed from the ideal pairs and also from the output pairs. All output pairs that have a first component that is not in the ideal set are removed.

There is one special case for the CV-system, which uses the Ideal Metric for the non-right-unique `instanceOf` relation. To allow for a fair comparison, we used the *Relaxed Ideal Metric*, which does not make the ideal pairs right-unique. The calculation of recall is relaxed as follows:

$$recall = \frac{|O \cap I|}{|\{x | \exists y : (x, y) \in I\}|}$$

Due to the effort, we could extract the ideal pairs only for a sub-corpus. To ensure significance in spite of this, we compute confidence intervals for our estimates: We interpret the sequence of output pairs as a repetition of a Bernoulli-experiment, where the output pair can be either correct (i.e. contained in the ideal pairs) or not. The parameter of this Bernoulli-distribution is the precision. We estimate the precision by drawing a sample (i.e. by extracting all ideal pairs in the sub-corpus). By assuming that the output pairs are identically independently distributed, we can calculate a confidence interval for our estimation. We report confidence intervals for precision and recall for a confidence level of $\alpha = 95\%$. We measure precision at different levels of recall and report the values for the best F1 value. We used approximate string matching techniques to account for different writings of the same entity. For example, we count the output pair ”Chopin” / ”composer” as correct, even if the ideal pairs contain ”Frederic_Chopin” / ”composer”. To ensure that LEILA does not just reproduce the example pairs, we list the percentage of examples among the output pairs. During our evaluation, we found that the Link Grammar parser does not finish parsing on roughly 1% of the files for unknown reasons.

[3] http://www.sourceforge.net/projects/texttoonto

[4] http://www.lonelyplanet.com/

Table 1: Results with different relations

Corpus	Relation	System	#D	#O	#C	#I	Precision	Recall	F1	%E
Wikicomposers	birthdate	LEILA(SVM)	87	95	70	101	73.68% ± 8.86%	69.31% ± 9.00%	71.43%	4.29%
Wikicomposers	birthdate	LEILA(kNN)	87	90	70	101	78.89% ± 8.43%	70.30% ± 8.91%	74.35%	4.23%
Wikigeography	synonymy	LEILA(SVM)	81	92	74	164	80.43% ± 8.11%	45.12% ± 7.62%	57.81%	5.41%
Wikigeography	synonymy	LEILA(kNN)	81	143	105	164	73.43% ± 7.24%	64.02% ± 7.35%	68.40%	4.76%
Wikicomposers	instanceOf	LEILA(SVM)	87	685	408	1127	59.56% ± 3.68%	36.20% ± 2.81%	45.03%	6.62%
Wikicomposers	instanceOf	LEILA(kNN)	87	790	463	1127	58.61% ± 3.43%	41.08% ± 2.87%	48.30%	7.34%
Wikigeneral	instanceOf	LEILA(SVM)	287	921	304	912	33.01% ± 3.04%	33.33% ± 3.06%	33.17%	3.62%
Googlecomposers	instanceOf	LEILA(SVM)	100	787	210	1334	26.68% ± 3.09%	15.74% ± 1.95%	19.80%	4.76%
Googlecomposers	instanceOf	LEILA(kNN)	100	840	237	1334	28.21% ± 3.04%	17.77% ± 2.05%	21.80%	8.44%
Googlec.+Wikic.	instanceOf	LEILA(SVM)	100	563	203	1334	36.06% ± 3.97%	15.22% ± 1.93%	21.40%	5.42%
Googlec.+Wikic.	instanceOf	LEILA(kNN)	100	826	246	1334	29.78% ± 3.12%	18.44% ± 2.08%	22.78%	7.72%

#O – number of output pairs
#C – number of correct output pairs
#I – number of ideal pairs

#D – number of documents in the hand-processed sub-corpus
%E – proportion of example pairs among the correct output pairs
Recall and Precision with confidence interval at $\alpha = 95\%$

3.2 Results

3.2.1 Results on different relations

Table 1 summarizes our experimental results with LEILA on different relations. For the **birthdate** relation, we used Edward Morykwas' list of famous birthdays[5] as examples. As counterexamples, we chose all pairs of a person that was in the examples and an incorrect birthdate. All pairs of a proper name and a date are candidates. We ran LEILA on the Wikicomposer corpus. LEILA performed quite well on this task. The patterns found were of the form ”*X* was born in *Y*” and ”*X* (*Y*)”.

For the **synonymy** relation we used all pairs of proper names that share the same synset in WordNet as examples (e.g. ”UN”/”United Nations”). As counterexamples, we chose all pairs of nouns that are not synonymous in WordNet (e.g. ”rabbit”/”composer”). All pairs of proper names are candidates. We ran LEILA on the Wikigeography corpus, because this set is particularly rich in synonyms. LEILA performed reasonably well. The patterns found include ”*X* was known as *Y*” as well as several non-grammatical constructions such as ”*X* (formerly *Y*)”.

For the **instanceOf** relation, it is difficult to select example pairs, because if an entity belongs to a concept, it also belongs to all super-concepts. However, admitting each pair of an entity and one of its super-concepts as an example would result in far too many false positives. The problem is to determine for each entity the (super-)concept that is most likely to be used in a natural language definition of that entity. Psychological evidence (Rosch et al., 1976) suggests that humans prefer a certain layer of concepts in the taxonomy to classify entities. The set of these concepts is called the *Basic Level*. Heuristically, we found that the lowest super-concept in WordNet that is not a compound word is a good approximation of the basic level concept for a given entity. We used all pairs of a proper name and the corresponding basic level concept of WordNet as examples. We could not use pairs of proper names and incorrect super-concepts as counterexamples, because our corpus Wikipedia knows more meanings of proper names than WordNet. Therefore, we used all pairs of a common noun and an incorrect super-concept from WordNet as counterexamples. All pairs of a proper name and a WordNet concept are candidates.

We ran LEILA on the Wikicomposers corpus. The performance on this task was acceptable, but not impressive. However, the chances to obtain a high recall and a high precision were significantly decreased by our tough evaluation policy: The ideal pairs include tuples deduced by resolving syntactic and semantic ambiguities and anaphoras. Furthermore, our evaluation policy demands that non-defining concepts like `member` not be chosen as instance concepts. In fact, a high proportion of the incorrect assignments were `friend`, `member`, `successor` and `predecessor`, decreasing the precision of LEILA. Thus, compared to the gold standard of humans, the performance of LEILA can be considered reasonably good. The patterns found include the Hearst patterns (Hearst, 1992) ”*Y* such as *X*”, but also more complex patterns like ”*X* was known as a *Y*”, ”*X* [. . .] as *Y*”, ”*X* [. . .] can be regarded as *Y*” and ”*X* is unusual among *Y*”. Some of these patterns could not have been found by primitive regular expression matching.

To test whether thematic heterogeneity influences LEILA, we ran it on the Wikigeneral corpus. Finally, to try the limits of our system, we ran it on the Googlecomposers corpus. As shown in Table 1, the performance of LEILA dropped in these increasingly challenging tasks, but LEILA could still produce useful results. We can improve the results on the Googlecomposers corpus by adding the Wikicomposers corpus for training.

[5] http://www.famousbirthdates.com

The different learning methods (kNN and SVM) performed similarly for all relations. Of course, in each of the cases, it is possible to achieve a higher precision at the price of a lower recall. The runtime of the system splits into parsing ($\approx 40s$ for each document, e.g. 3:45h for Wikigeography) and the core algorithm (2-15min for each corpus, 5h for the huge Wikigeneral).

3.2.2 Results with different competitors

Table 2 shows the results for comparing LEILA against various competitors (with LEILA in boldface). We compared LEILA to **TextToOnto** and **Text2Onto** for the `instanceOf` relation on the Wikicomposers corpus. TextToOnto requires an ontology as source of possible concepts. We gave it the WordNet ontology, so that it had the same preconditions as LEILA. Text2Onto does not require any input. Text2Onto seems to have a precision comparable to ours, although the small number of found pairs does not allow a significant conclusion. Both systems have drastically lower recall than LEILA.

For **Snowball**, we only had access to the test corpus. Hence we trained LEILA on a small portion (3%) of the test documents and tested on the remaining ones. Since the original 5 seed pairs that Snowball used did not appear in the collection at our disposal, we chose 5 other pairs as examples. We used no counterexamples and hence omitted the Training Phase of our algorithm. LEILA quickly finds the pattern ”*Y*-based *X*”. This led to very high precision and good recall, compared to Snowball – even though Snowball was trained on a much larger training collection.

The **CV-system** differs from LEILA, because its ideal pairs are a table, in which each entity is assigned to its most likely concept according to a human understanding of the text – independently of whether there are explicit definitions for the entity in the text or not. We conducted two experiments: First, we used the document set used in Cimiano and Völker’s original paper (Cimiano and Völker, 2005a), the Lonely Planet corpus. To ensure a fair comparison, we trained LEILA separately on the Wikicomposers corpus, so that LEILA cannot have example pairs in its output. For the evaluation, we calculated precision and recall with respect to an ideal table provided by the authors. Since the CV-system uses a different ontology, we allowed a distance of 4 edges in the WordNet hierarchy to count as a match (for both systems). Since the explicit definitions that our system relies on were sparse in the corpus, LEILA performed worse than the competitor. In a second experiment, we had the CV-system run on the Wikicomposers corpus. As the CV-system requires a set of target concepts, we gave it the set of all concepts in our ideal pairs. Furthermore, the system requires an ontology on these concepts. We gave it the WordNet ontology, pruned to the target concepts with their super-concepts. We evaluated by the Relaxed Ideal Metric, again allowing a distance of 4 edges in the WordNet hierarchy to count as a match (for both systems). This time, our competitor performed worse. This is because our ideal table is constructed from the definitions in the text, which our competitor is not designed to follow. These experiments only serve to show the different philosophies in the definition of the ideal pairs for the CV-system and LEILA. The CV-system does not depend on explicit definitions, but it is restricted to the `instanceOf`-relation.

4 Conclusion and Outlook

We addressed the problem of automatically extracting instances of arbitrary binary relations from natural language text. The key novelty of our approach is to apply a deep syntactic analysis to this problem. We have implemented our approach and showed that our system LEILA outperforms existing competitors.

Our current implementation leaves room for future work. For example, the linkages allow for more sophisticated ways of resolving anaphoras or matching patterns. LEILA could learn numerous interesting relations (e.g. `country` / `president` or `isAuthorOf`) and build up an ontology from the results with high confidence. LEILA could acquire and exploit new corpora on its own (e.g., it could read newspapers) and it could use its knowledge to acquire and structure its new knowledge more efficiently. We plan to exploit these possibilities in our future work.

4.1 Acknowledgements

We would like to thank Eugene Agichtein for his caring support with Snowball. Furthermore, Johanna Völker and Philipp Cimiano deserve our sincere thanks for their unreserved assistance with their system.

References

[Agichtein and Gravano2000] E. Agichtein and L. Gravano. 2000. *Snowball*: extracting relations from large plain-text collections. In *ACM 2000*, pages 85–94, Texas, USA.

[Brin1999] Sergey Brin. 1999. Extracting patterns and relations from the world wide web. In *Selected papers from the Int. Workshop on the WWW and Databases*, pages 172–183, London, UK. Springer-Verlag.

[Buitelaar and Ramaka2005] P. Buitelaar and S. Ramaka. 2005. Unsupervised ontology-based semantic tagging

Table 2: Results with different competitors

Corpus	M	Relation	System	#D	#O	#C	#I	Prec	Rec	F1
Snowball corp.	S	headquarters	LEILA(SVM)	54	92	82	165	**89.13%**± 6.36%	**49.70%**± 7.63%	**63.81%**
Snowball corp.	S	headquarters	LEILA(kNN)	54	91	82	165	**90.11%**± 6.13%	**49.70%**± 7.63%	**64.06%**
Snowball corp.	S	headquarters	Snowball	54	144	49	165	34.03%± 7.74%	29.70%± 6.97%	31.72%
Snowball corp.	I	headquarters	LEILA(SVM)	54	50	48	126	**96.00%**± 5.43%	**38.10%**± 8.48%	**54.55%**
Snowball corp.	I	headquarters	LEILA(kNN)	54	49	48	126	**97.96%**± 3.96%	**38.10%**± 8.48%	**54.86%**
Snowball corp.	I	headquarters	Snowball	54	64	31	126	48.44%±12.24%	24.60%± 7.52%	32.63%
Wikicomposers	S	instanceOf	LEILA(SVM)	87	685	408	1127	**59.56%**± 3.68%	**36.20%**± 2.81%	**45.03%**
Wikicomposers	S	instanceOf	LEILA(kNN)	87	790	463	1127	**58.61%**± 3.43%	**41.08%**± 2.87%	**48.30%**
Wikicomposers	S	instanceOf	Text2Onto	87	36	18	1127	50.00%	1.60%± 0.73%	3.10%
Wikicomposers	S	instanceOf	TextToOnto	87	121	47	1127	38.84%± 8.68%	4.17%± 1.17%	7.53%
Wikicomposers	R	instanceOf	LEILA(SVM)	87	336	257	744	**76.49%**± 4.53%	**34.54%**± 3.42%	**47.59%**
Wikicomposers	R	instanceOf	LEILA(kNN)	87	367	276	744	**75.20%**± 4.42%	**37.10%**± 3.47%	**49.68%**
Wikicomposers	R	instanceOf	CV-system	87	134	30	744	22.39%	4.03%± 1.41%	6.83%
Lonely Planet	R	instanceOf	LEILA(SVM)	–	159	42	289	**26.42%**± 6.85%	**14.53%**± 4.06%	**18.75%**
Lonely Planet	R	instanceOf	LEILA(kNN)	–	168	44	289	**26.19%**± 6.65%	**15.22%**± 4.14%	**19.26%**
Lonely Planet	R	instanceOf	CV-system	–	289	92	289	31.83%± 5.37%	31.83%± 5.37%	31.83%

M – Metric (S: Standard, I: Ideal Metric, R: Relaxed Ideal Metric). Other abbreviations as in Table 1

for knowledge markup. In W. Buntine, A. Hotho, and Stephan Bloehdorn, editors, *Workshop on Learning in Web Search at the ICML 2005*.

[Buitelaar et al.2004] P. Buitelaar, D. Olejnik, and M. Sintek. 2004. A protege plug-in for ontology extraction from text based on linguistic analysis. In *ESWS 2004*, Heraklion, Greece.

[Califf and Mooney1997] M. Califf and R. Mooney. 1997. Relational learning of pattern-match rules for information extraction. *ACL-97 Workshop in Natural Language Learning*, pages 9–15.

[Cimiano and Völker2005a] P. Cimiano and J. Völker. 2005a. Text2onto - a framework for ontology learning and data-driven change discovery. In A. Montoyo, R. Munozand, and E. Metais, editors, *Proc. of the 10th Int. Conf. on Applications of Natural Language to Information Systems*, pages 227–238, Alicante, Spain.

[Cimiano and Völker2005b] P. Cimiano and J. Völker. 2005b. Towards large-scale, open-domain and ontology-based named entity classification. In *Int. Conf. on Recent Advances in NLP 2005*, pages 166–172.

[Cimiano et al.2005] P. Cimiano, G. Ladwig, and S. Staab. 2005. Gimme the context: Contextdriven automatic semantic annotation with cpankow. In Allan Ellis and Tatsuya Hagino, editors, *WWW 2005*, Chiba, Japan.

[Etzioni et al.2004] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. 2004. Web-scale information extraction in knowitall (preliminary results). In *WWW 2004*, pages 100–110.

[Fellbaum1998] C. Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

[Finn and Kushmerick2004] A. Finn and N. Kushmerick. 2004. Multi-level boundary classification for information extraction. In *ECML 2004*, pages 111–122.

[Freitag and Kushmerick2000] D. Freitag and N. Kushmerick. 2000. Boosted wrapper induction. In *American Nat. Conf. on AI 2000*.

[Graupmann2004] Jens Graupmann. 2004. Concept-based search on semi-structured data exploiting mined semantic relations. In *EDBT Workshops*, pages 34–43.

[Hearst1992] A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *ICCL 1992*, Nantes, France.

[J. Iria2005] F. Ciravegna J. Iria. 2005. Relation extraction for mining the semantic web.

[Maedche and Staab2000] A. Maedche and S. Staab. 2000. Discovering conceptual relations from text. In W. Horn, editor, *ECAI 2000*, pages 85–94, Berlin, Germany.

[Mann and Yarowsky2005] Gideon Mann and David Yarowsky. 2005. Multi-field information extraction and cross-document fusion. In *ACL 2005*.

[Montague1974] R. Montague. 1974. Universal grammar. In *Formal Philosophy. Selected Papers of Richard Montague.* Yale University Press.

[Ravichandran and Hovy2002] D. Ravichandran and E. Hovy. 2002. Learning surface text patterns for a question answering system. In *ACL 2002*, Philadelphia, USA.

[Riloff1996] E. Riloff. 1996. Automatically generating extraction patterns from untagged text. *Annual Conf. on AI 1996*, pages 1044–1049.

[Rosch et al.1976] E. Rosch, C.B. Mervis, W.D. Gray, D.M. Johnson, and P. Boyes-Bream. 1976. Basic objects in natural categories. *Cognitive Psychology*, pages 382–439.

[Ruiz-Casado et al.2005] Maria Ruiz-Casado, Enrique Alfonseca, and Pablo Castells. 2005. Automatic extraction of semantic relationships for wordnet by means of pattern learning from wikipedia. In *NLDB 2006*, pages 67–79.

[Sleator and Temperley1993] D. Sleator and D. Temperley. 1993. Parsing english with a link grammar. *3rd Int. Workshop on Parsing Technologies*.

[Soderland et al.1995] S. Soderland, D. Fisher, J. Aseltine, and W. Lehnert. 1995. Crystal: Inducing a conceptual dictionary. *IJCAI 1995*, pages 1314–1319.

[Soderland1999] S. Soderland. 1999. Learning information extraction rules for semi-structured and free text. *Machine Learning*, pages 233–272.

[Suchanek et al.2006] Fabian M. Suchanek, Georgiana Ifrim, and Gerhard Weikum. 2006. Combining Linguistic and Statistical Analysis to Extract Relations from Web Documents. In *SIGKDD 2006*.

[Xu and Krieger2003] F. Xu and H. U. Krieger. 2003. Integrating shallow and deep nlp for information extraction. In *RANLP 2003*, Borovets, Bulgaria.

[Xu et al.2002] F. Xu, D. Kurz, J. Piskorski, and S. Schmeier. 2002. Term extraction and mining term relations from free-text documents in the financial domain. In *Int. Conf. on Business Information Systems 2002*, Poznan, Poland.

[Yangarber et al.2000] R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen. 2000. Automatic acquisition of domain knowledge for information extraction. In *ICCL 2000*, pages 940–946, Morristown, NJ, USA. Association for Computational Linguistics.

[Yangarber et al.2002] R. Yangarber, W. Lin, and R. Grishman. 2002. Unsupervised learning of generalized names. In *ICCL 2002*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics.

Ontology Population from Textual Mentions: Task Definition and Benchmark

Bernardo Magnini, Emanuele Pianta, Octavian Popescu and Manuela Speranza

ITC-irst, Istituto per la Ricerca Scientifica e Tecnologica
Via Sommarive 18, 38050 Povo (TN), Italy
{magnini, pianta, popescu, manspera}@itc.it

Abstract

In this paper we propose and investigate Ontology Population from Textual Mentions (OPTM), a sub-task of Ontology Population from text where we assume that mentions for several kinds of entities (e.g. PERSON, ORGANIZATION, LOCATION, GEO-POLITICAL_ ENTITY) are already extracted from a document collection. On the one hand, OPTM simplifies the general Ontology Population task, limiting the input textual material; on the other hand, it introduces challenging extensions to Ontology Population restricted to named entities, being open to a wider spectrum of linguistic phenomena. We describe a manually created benchmark for OPTM and discuss several factors which determine the difficulty of the task.

1 Introduction

Mentions are portions of text which refer to entities[1]. As an example, given a particular textual context, both the mentions "*George W. Bush*" and "*the U.S. President.*" refer to the same entity, i.e. a particular instance of Person whose first name is "*George*", whose middle initial is "*W.*", whose family name is "*Bush*" and whose role is "*U.S. President*".

In this paper we propose and investigate Ontology Population from Textual Mentions (OPTM), a sub-task of Ontology Learning and Population (OLP) from text where we assume that mentions for several kinds of entities (e.g. PERSON, ORGANIZATION, LOCATION, GEO-POLITICAL _ENTITY) are already extracted from a document collection.

We assume an ontology with a set of classes $C=\{c_1, ..., c_n\}$ with each class c_1 being described by a set of attribute value pairs $[a_1, v_1]$. Given a set of mentions $M=\{m_{1,c1}, ..., m_{n,cn}\}$, where each mention m_j is classified into a class c_i in C, the OPTM task is defined in three steps: Recognition and Classification of Entity Attributes, Normalization, and Resolution of inter-text Entity Co-reference.

(i) **Recognition and Classification of Entity Attributes (RCEA)**. The textual material expressed in a mention is extracted and distributed along the attribute-value pairs already defined for the class c_i of the mention; as an example, given the PERSON mention "*U.S. President Bush*", we expect that the attribute LAST_NAME is filled with the value "*Bush*" and the attribute ROLE is filled with the value "*U.S. President*". Note that fillers, at this step, are still portions of text.

(ii) **Normalization**. The textual material extracted at step (i) is assigned to concepts and relations already defined in the ontology; for example, the entity BUSH is created as an instance of COUNTRY_PRESIDENT, and an instance of the relation PRESIDENT_OF is created between BUSH and U.S.A. At this step different instances are created for co-referring mentions.

(iii) **Resolution of inter-text Entity Co-reference (REC)**. Each mention m_j has to be assigned to a single individual entity belonging to a class in C. For example, we recognize that the instances created at step (i) for "*U.S. President Bush*" and "*George W. Bush*" actually refer to the same entity.

[1] The terms "mention" and "entity" have been introduced within the ACE Program (Linguistic Data Consortium, 2004). "Mentions" are equivalent to "referring expressions" and "entities" are equivalent to "referents", as widely used in computational linguistics. In this paper, we use italics for "*mentions*" and small caps for ENTITY and ENTITY_ATTRIBUTE.

Proceedings of the 2nd Workshop on Ontology Learning and Population, pages 26–32,
Sydney, July 2006. ©2006 Association for Computational Linguistics

In this paper we address steps (i) and (iii), while step (ii) is work in progress. The input of the OPTM task consists of classified mentions and the output consists of individual entities filled with textual material (i.e. there is no normalization) with their co-reference relations. The focus is on the definition of the task and on an empirical analysis of the aspects that determine its complexity, rather than on approaches and methods for the automatic solution of OPTM.

There are several advantages of OPTM which make it appealing for OLP. First, mentions provide an obvious simplification with respect to the more general Ontology Population from text (cf. Buitelaar et al. 2005); in particular, mentions are well defined and there are systems for automatic mention recognition. Although there is no univocally accepted definition for the OP task, a useful approximation has been suggested by (Bontcheva and Cunningham, 2005) as Ontology Driven Information Extraction with the goal of extracting and classifying instances of concepts and relations defined in a Ontology, in place of filling a template. A similar task has been approached in a variety of perspectives, including term clustering (Lin, 1998 and Almuhareb and Poesio, 2004) and term categorization (Avancini et al. 2003). A rather different task is Ontology Learning, where new concepts and relations are supposed to be acquired, with the consequence of changing the definition of the Ontology itself (Velardi et al. 2005). However, since mentions have been introduced as an evolution of the traditional Named Entity Recognition task (see Tanev and Magnini, 2006), they guarantee a reasonable level of difficulty, which makes OPTM challenging both for the Computational Linguistic side and the Knowledge Representation community. Second, there already exist annotated data with mentions, delivered under the ACE (Automatic Content Extraction) initiative (Ferro et al. 2005, Linguistic Data Consortium 2004), which makes the exploitation of machine learning based approaches possible. Finally, having a limited scope with respect to OLP, the OPTM task allows for a better estimation of performance; in particular, it is possible to evaluate more easily the recall of the task, i.e. the proportion of information correctly assigned to an entity out of the total amount of information provided by a certain mention.

In the paper we both define the OPTM task and describe an OPTM benchmark, i.e. a document collection annotated with mentions as well as an ontology where information from mentions has been manually extracted. The general architecture of the OPTM task has been sketched above, considering three sub tasks. The document collection we use consists of about 500 Italian news items. Currently, mentions referring to PERSON, ORGANIZATION and GEO-POLITICAL_ ENTITY have been annotated and co-references among such mentions have been established. As for the RCEA sub task, we have considered mentions referring to PERSON and have built a knowledge base of instances, each described with a number of attribute-value pairs.

The paper is structured as follows. Section 2 provides the useful background as far as mentions and entities are concerned. Section 3 defines the OPTM task and introduces the dataset we have used, as well as the annotation procedures and guidelines we have defined for the realization of the OPTM benchmark corpus. Section 4 reports on a number of quantitative and qualitative analyses of the OPTM benchmark aimed at determining the difficulty of the task. Finally, Section 5 proposes future extensions and developments of our work.

2 Mentions and Entities

As indicated in the ACE Entity Detection task, the annotation of entities (e.g. PERSON, ORGANIZATION, LOCATION and GEO-POLITICAL_ENTITY) requires that the entities mentioned in a text be detected, their syntactic head marked, their sense disambiguated, and that selected attributes of these entities be extracted and merged into a unified representation for each entity.

As it often happens that the same entity is mentioned more than once in the same text, two inter-connected levels of annotation have been defined: the level of the entity, which provides a representation of an object in the world, and the level of the entity mention, which provides information about the textual references to that object. For instance, if the entity GEORGE_W._BUSH (e.g. the individual in the world who is the current president of the U.S.) is mentioned in two different sentences of a text as "*the U.S. president*" and as "*the president*", these two expressions are considered as two co-referring entity mentions.

The kinds of reference made by entities to something in the world are described by the following four classes:

- **specific referential entities** are those where the entity being referred to is a unique object or set of objects (e.g. "*The president of thecompany* is here*"*)[2];

- **generic referential entities** refer to a kind or type of entity and not to a particular object (or set of objects) in the world (e.g. "*The president* is elected every 5 years");
- **under-specified referential entities** are non-generic non-specific references, including imprecise quantifications (e.g. "*everyone*") and estimates (e.g. "*more than 10.000 people*");
- **negatively quantified entities** refer to the empty set of the mentioned type of object (e.g. "*No lawyer*").

The textual extent of mentions is defined as the entire nominal phrase used to refer to an entity, thus including modifiers (e.g. "*a big family*"), prepositional phrases (e.g. "*the President of the Republic*") and dependent clauses (e.g. "*the girl who is working in the garden*").

The classification of entity mentions is based on syntactic features; among the most significant categories defined by LDD (Linguistic Data Consortium 2004) there are:

- NAM: proper names (e.g. "*Ciampi*", "*the UN*");
- NOM: nominal constructions (e.g. "*good children*"*,* "*the company*");
- PRO: pronouns, e.g. personal ("*you*") and indefinite ("*someone*");
- WHQ: wh-words, such as relatives and interrogatives (e.g. "*Who's there?*");
- PTV: partitive constructions (e.g. "*some of them*"*,* "*one of the schools*");
- APP: appositive constructions (e.g. "*Dante, famous poet*", "*Juventus, Italian football club*"*).*

Since the dataset presented in this paper has been developed for Italian, some new types of mentions have been added to those listed in the LDC guidelines; for instance, we have created a specific tag, ENCLIT, to annotate the clitics whose extension can not be identified at word-level (e.g. "*veder*[*lo*]"/"to see *him*"). Some types of mentions, on the other hand, have been eliminated; this is the case for pre-modifiers, due to syntactic differences between English, where both adjectives and nouns can be used as pre-modifiers, and Italian, which only admits adjectives in that position.

In extending the annotation guidelines, we have decided to annotate all conjunctions of entities, not only those which share the same modifiers as indicated in the ACE guidelines, and to mark them using a specific new tag, CONJ (e.g. "*mother and child*"*)*[3].

According to the ACE standards, each distinct person or set of people mentioned in a document refers to an entity of type PERSON. For example, people may be specified by name ("*John Smith*"), occupation ("*the butcher*"), family relation ("*dad*"), pronoun ("*he*"), etc., or by some combination of these.

PERSON (PE), the class we have considered for the Ontology Population from Textual Mention task, is further classified with the following subtypes:

- INDIVIDUAL_PERSON: PEs which refer to a single person (e.g. "*George W. Bush"*);
- GROUP_PERSON: PEs which refer to more than one person (e.g. "*my parents", "your family"*, etc.);
- INDEFINITE_PERSON: a PE is classified as indefinite when it is not possible to judge from the context whether it refers to one or more persons (e.g. "I wonder *who* came to see me*"*).

3 Task definition

In Section 3.1 we first describe the document collection we have used for the creation of the OPTM benchmark. Then, Section 3.2 provides details about RCEA, the first step in OPTM.

3.1 Document collection

The OPTM benchmark is built on top of a document collection (I-CAB, Italian Content Annotated Bank)[4] annotated with entity mentions. I-CAB (Magnini et al. 2006) consists of 525 news documents taken from the local newspaper 'L'Adige'[5]. The selected news stories belong to four different days (September, 7th and 8th 2004 and October, 7th and 8th 2004) and are grouped into five categories: News Stories, Cultural News, Economic News, Sports News and Local News (see Table 1).

	09/07	09/08	10/07	10/08	Total
News	23	25	18	21	87
Culture	20	18	16	18	72
Economy	13	15	12	14	54
Sport	29	41	27	26	123
Local	46	43	49	51	189
TOTAL	131	142	122	130	525

Table 1: Number of news stories per category.

[2] Notice that the corpus is in Italian, but we present English examples for the sake of readability.

[3] Appositive and conjoined mentions are complex constructions. Although LDC does not identify heads for complex constructions, we have decided to annotate all the extent as head.

[4] A demo is available at http://ontotext.itc.it/webicab

[5] http://www.ladige.it/

I-CAB is further divided into training and test sections, which contain 335 and 190 documents respectively. In total, I-CAB consists of around 182,500 words: 113,500 and 69,000 words in the training and the test sections respectively (the average length of a news story is around 339 words in the training section and 363 words in the test section).

The annotation of I-CAB is being carried out manually, as we intend I-CAB to become a benchmark for various automatic Information Extraction tasks, including recognition and normalization of temporal expressions, entities, and relations between entities (e.g. the relation affiliation connecting a person to the organization to which he or she is affiliated).

3.2 Recognition and Classification

As stated in Section 1, we assume that for each type of entity there is a set of attribute-value pairs, which typically are used for mentioning that entity type. The same entity may have different values for the same attribute and, at this point no normalization of the data is made, so there is no way to differentiate between different values of the same attribute, e.g. there is no stipulation regarding the relationship between "*politician*" and "*political leader*". Finally, we currently assume a totally flat structure among the possible values for the attributes.

The work we describe in this Section and in the next one concerns a pilot study on entities of type PERSON. After an empirical investigation on the dataset described in Section 3.1 we have assumed that the attributes listed in the first column of Table 2 constitute a proper set for this type of entity. The second column lists some possible values for each attribute.

The textual extent of a value is defined as the maximal extent containing pertinent information. For instance, if we have a person mentioned as "*the thirty-year-old sport journalist*", we will select "*sport journalist*" as value for the attribute ACTIVITY. In fact, the age of the journalist in not pertinent to the activity attribute and is left out, whereas "*sport*" contributes to specifying the activity performed.

As there are always less paradigmatic values for a given attribute, we shortly present further the guidelines in making a decision in those cases. Generally, articles and prepositions are not admitted at the beginning of the textual extent of a value, an exception being made in the case of articles in nicknames.

Attributes	Possible values
FIRST_NAME	*Ralph, Greg*
MIDDLE_NAME	*J., W.*
LAST_NAME	*McCarthy, Newton*
NICKNAME	*Spider, Enigmista*
TITLE	*prof., Mr.*
SEX	*actress*
ACTIVITY	*journalist, doctor*
AFFILIATION	*The New York Times*
ROLE	*director, president*
PROVENIENCE	*South American*
FAMILY_RELATION	*father, cousin*
AGE_CATEGORY	*boy, girl*
MISCELLANEA	*The men with red shoes*

Table 2. Attributes for PERSON.

Typical examples for the TITLE attribute are "*Mister*", "*Miss*", "*Professor*", etc. We consider as TITLE the words which are used to address people with special status, but which do not refer specifically to their activity. In Italian, professions are often used to address people (e.g. "*avvocato/lawyer*", "*ingegnere/engineer*"). In order to avoid a possible overlapping between the TITLE attribute and the ACTIVITY attribute, professions are considered values for title only if they appear in abbreviated forms ("*avv.*", "*ing.*" etc.) before a proper name.

With respect to the SEX attribute, we consider as values all the portions of text carrying this information. In most cases, first and middle names are relevant. In addition, the values of the SEX attribute can be gendered words (e.g. "*Mister*" vs. "*Mrs.*", "*husband*" vs. "*wife*") and words from grammatical categories carrying information about gender (e.g. adjectives).

The attributes ACTIVITY, ROLE, AFFILIATION are three strictly connected attributes. ACTIVITY refers to the actual activity performed by a person, while ROLE refers to the position they occupy. So, for instance, "*politician*" is a possible value for ACTIVITY, while "*leader of the Labour Party*" refers to a ROLE. Each group of these three attributes is associated with a mention and all the information within a group has to be derived from the same mention. If different pieces of information derive from distinct mentions, we will have two separate groups. Consider the following three mentions of the same entity:

(1) "the journalist of Radio Liberty"
(2) "the redactor of breaking news"
(3) "a spare time astronomer"

These three mentions lead to three different groups of ACTIVITY, ROLE and AFFILIATION. The obvious inference that the first two mentions conceptually belong to the same group is not drawn. This step is to be taken at a further stage.

The PROVENIENCE attribute can have as values all phrases denoting geographical/racial origin or provenience and religious affiliation.

The attribute AGE_CATEGORY can have either numerical values, such as "*three years old*", or words indicating age, such as "*middle-aged*", etc.

In the next section we will analyze the occurrences of the values of these attributes in a news corpus.

4 Data analysis

The difficulty of the OPTM task is directly correlated to four factors: (i) the extent to which the linguistic form of mentions varies; (ii) the perplexity of the values of the attributes; (iii) the size of the set of the potential co-references and (iv) the number of different mentions per entity. In this section we present the work we have undertaken so far and the results we have obtained regarding the above four factors.

We started with a set of 175 documents belonging to the I-CAB corpus (see Section 3.1). Each document has been manually annotated observing the specifications described in Section 3.2. We focused on mentions referring to INDIVIDUAL PERSON (Mentions in Table 3), excluding from the dataset both mentions referring to different entity types (e.g. ORGANIZATION) and PERSON GROUP. In addition, for the purposes of this work we decided to filter out the following mentions: (i) mentions consisting of a single pronoun; (ii) nested mentions, (in particular in the case where a larger mention, e.g. "*President Ciampi*", contained a smaller one, e.g. "*Ciampi*", only the larger mention was considered). The total number of remaining mentions (Meaningful mentions in Table 3) is 2343. Finally, we filtered out repetitions of mentions (i.e. string equal) that co-refer inside the same document, obtaining a set of 1139 distinct mentions.

The average number of mentions for an entity in a document is 2.09, while the mentions/entity proportion within the whole collection is 2.68. The detailed distribution of mentions with respect to document entities is presented in Table 4. Columns 1 and 3 list the number of mentions and columns 2 and 4 list the number of entities which are mentioned for the respective number of times (from 1 to 9 and more than 10). For instance, in the dataset there are 741 entities which, within a single document, have just one mention, while there are 27 entities which are mentioned more than 10 times in the same document. As an indication of variability, only 14% of document entities have been mentioned in two different ways.

Documents	175
Words	57 033
Words in mentions	8116
Mentions	3157
Meaningful mentions	2343
Distinct mentions	1139
Document entities	1117
Collection entities	873

Table 3. Documents, mentions and entities in the OPTM dataset.

#M/E	#occ	#M/E	#occ
1	741	6	15
2	164	7	11
3	64	8	12
4	47	9	5
5	31	≥10	27

Table 4. Distribution of mentions per entity.

4.1 Co-reference density

We can estimate the a priori probability that two entities selected from different documents co-refer. Actually, this is the estimate of the probability that two entities co-refer conditioned by the fact that they have been correctly identified inside the documents. We can compute such probability as the complement of the ratio between the number of different entities and the number of the document entities in the collection.

$$P(cross - coref) = 1 - \frac{\#collection - entities}{\#document - entities}$$

From Table 3 we read these values as 873 and 1117 respectively, therefore, for this corpus, the probability of intra-document co-reference is approximately 0.22.

A cumulative factor in estimating the difficulty of the co-reference task is the ratio between the number of different entities and the number of mentions. We call this ratio the *co-reference density* and it shows the a priori expectation that a correct identified mention refers to a new entity.

$$coref - density = \frac{\#collection - entities}{\#mentions}$$

The co-reference density takes values in the interval with limits [0-1]. The case where the co-reference density tends to 0 means that all the mentions refer to the same entity, while where the value tends to 1 it means that each mention in the collection refers to a different entity. Both limits render the co-reference task superfluous. The figure for co-reference density we found in our corpus is 873/2343 ≈ 0.37, and it is far from being close to one of the extremes.

A last measure we introduce is the ratio between the number of different entities and the number of distinct mentions. Let's call it *pseudo co-reference density*. In fact it shows the value of co-reference density conditioned by the fact that one knows in advance whether two mentions that are identical also co-refer.

$$pcoref - density = \frac{\#collection - entities}{\#distinct - mentions}$$

The pseudo co-reference for our corpus is 873/1139 ≈ 0.76. This information is not directly expressed in the collection, so it should be approximated. The difference between co-reference density and pseudo co-reference density (see Table 5) shows the increase in recall, if one considers that two identical mentions refer to the same entity with probability 1. On the other hand, the loss in accuracy might be too large (consider for example the case when two different people happen to have the same first name).

co-reference density	0.37
pseudo co-reference density	0.76
cross co-reference	0.22

Table 5. A priori estimation of difficulty of co-reference

4.2 Attribute variability

The estimation of the variability of the values for a certain attribute is given in Table 6. The first column indicates the attribute under consideration; the second column lists the total number of mentions of the attribute found in the corpus; the third column lists the number of different values that the attribute actually takes and, between parentheses, its proportion over the total number of values; the fourth column indicates the proportion of the occurrences of the attribute with respect to the total number of mentions (distinct mentions are considered).

Attributes	**total occ.**	**distinct occ. (%)**	**occ. prob.**
FIRST_NAME	535	303 (44%)	27,0%
MIDDLE_NAME	25	25 (100%)	2,1%
LAST_NAME	772	690 (11%)	61,0%
NICKNAME	14	14 (100%)	1,2%
TITLE	12	10 (17%)	0,8%
SEX	795	573 (23%)	51,0%
ACTIVITY	145	88 (40%)	7,0%
AFFILIATION	134	121 (10%)	11,0%
ROLE	155	92 (42%)	8,0%
PROVENIENCE	120	80 (34%)	7,3%
FAMILY_REL.	17	17(100%)	1,4%
AGE_CATEGORY	31	31(100%)	2,7%
MISCELLANEA	106	106 (100%)	9,3%

Table 6. Variability of values for attributes.

In Table 7 we show the distribution of the attributes inside one mention. That is, we calculate how many times one entity contains more than one attribute. Columns 1 and 3 list the number of attributes found in a mention, and columns 2 and 4 list the number of mentions that actually contain that number of values for attributes.

#attributes	#mentions	#attributes	#mentions
1	398	5	55
2	220	6	25
3	312	7	8
4	117	8	4

Table 7. Number of attributes inside a mention.

An example of a mention from our dataset that includes values for eight attributes is the following:

The correspondent of Al Jazira, Amr Abdel Hamid, an Egyptian of Russian nationality…

We conclude this section with a statistic regarding the coverage of attributes (miscellanea excluded). There are 7275 words used in 1139 distinct mentions, out of which 3606, approximately 49%, are included in the values of the attributes.

5 Conclusion and future work

We have presented work in progress aiming at a better definition of the general OLP task. In particular we have introduced Ontology Population from Textual Mentions (OPTM) as a simplification of OLP, where the source textual material are already classified mentions of entities.
An analysis of the data has been conducted over a OPTM benchmark manually built from a corpus of Italian news. As a result a number of indicators have been extracted that suggest the complexity of the task for systems aiming at automatic resolution of OPTM.

Our future work is related to the definition and extension of the OPTM benchmark for the normalization step (see Introduction). For this step it is crucial the construction and use of a large-scale ontology, including the concepts and relations referred by mentions. A number of interesting relations between mentions and ontology are likely to emerge.

The work presented in this paper is part of the ONTOTEXT project, a larger initiative aimed at developing text mining technologies to be exploited in the perspective of the Semantic Web. The project focuses on the study and development of innovative knowledge extraction techniques for producing new or less noisy information to be made available to the Semantic Web. ONTOTEXT addresses three key research aspects: annotating documents with semantic and relational information, providing an adequate degree of interoperability of such relational information, and updating and extending the ontologies used for Semantic Web annotation. The concrete evaluation scenario in which algorithms will be tested with a number of large-scale experiments is the automatic acquisition of information about people from newspaper articles.

6 Acknowledgements

This work was partially funded the three-year project ONTOTEXT[6] funded by the Provincia Autonoma di Trento. We would like to thank Nicola Tovazzi for his contribution to the annotation of the dataset.

[6] http://tcc.itc.it/projects/ontotext

References

Almuhareb, A. and Poesio, M.. 2004. Attribute-based and value-based clustering: An evaluation. In Proceedings of EMNLP 2004, pages 158--165, Barcelona, Spain.

Avancini, H., Lavelli, A., Magnini, B., Sebastiani, F., Zanoli, R. (2003). Expanding Domain-Specific Lexicons by Term Categorization. In: Proceedings of SAC 2003, 793-79.

Cunningham, H. and Bontcheva, K. Knowledge Management and Human Language: Crossing the Chasm. Journal of Knowledge Management, 9(5), 2005.

Buitelaar, P., Cimiano, P. and Magnini, B. (Eds.) Ontology Learning from Text: Methods, Evaluation and Applications. IOS Press, 2005.

Ferro, L., Gerber, L., Mani, I., Sundheim, B. and Wilson, G. (2005). TIDES 2005 Standard for the Annotation of Temporal Expressions. Technical report, MITRE.

Lavelli, A., Magnini, B., Negri, M., Pianta, E., Speranza, M. and Sprugnoli, R. (2005). Italian Content Annotation Bank (I-CAB): Temporal Expressions (V. 1.0.). Technical Report T-0505-12. ITC-irst, Trento.

Lin, D. (1998). Automatic Retrieval and Clustering of Similar Words. In: Proceedings of COLING-ACL98, Montreal, Canada, 1998.

Linguistic Data Consortium (2004). ACE (Automatic Content Extraction) English Annotation Guidelines for Entities, version 5.6.1 2005.05.23. http://projects.ldc.upenn.edu/ace/docs/English-Entities-Guidelines_v5.6.1.pdf

Magnini, B., Pianta, E., Girardi, C., Negri, M., Romano, L., Speranza, M., Bartalesi Lenzi, V. and Sprugnoli, R. (2006). I-CAB: the Italian Content Annotation Bank. Proceedings of LREC-2006, Genova, Italy, 22-28 May, 2006.

Tanev, H. and Magnini, B. Weakly Supervised Approaches for Ontology Population. Proceedings of EACL-2006, Trento, Italy, 3-7 April, 2006.

Velardi, P., Navigli, R., Cuchiarelli, A., Neri, F. (2004). Evaluation of Ontolearn, a Methodology for Automatic Population of Domain Ontologies. In: Buitelaar, P., Cimiano, P., Magnini, B. (eds.): Ontology Learning from Text: Methods, Evaluation and Applications, IOS Press, Amsterdam, 2005.

Efficient Hierarchical Entity Classifier Using Conditional Random Fields

Koen Deschacht
Interdisciplinary Centre for Law & IT
Katholieke Universiteit Leuven
Tiensestraat 41, 3000 Leuven, Belgium
koen.deschacht@law.kuleuven.ac.be

Marie-Francine Moens
Interdisciplinary Centre for Law & IT
Katholieke Universiteit Leuven
Tiensestraat 41, 3000 Leuven, Belgium
marie-france.moens@law.kuleuven.be

Abstract

In this paper we develop an automatic classifier for a very large set of labels, the WordNet synsets. We employ Conditional Random Fields (CRFs) because of their flexibility to include a wide variety of non-independent features. Training CRFs on a big number of labels proved a problem because of the large training cost. By taking into account the hypernym/hyponym relation between synsets in WordNet, we reduced the complexity of training from $O(TM^2NG)$ to $O(T\,(logM)^2NG)$ with only a limited loss in accuracy.

1 Introduction

The work described in this paper was carried out during the CLASS project[1]. The central objective of this project is to develop advanced learning methods that allow images, video and associated text to be analyzed and structured automatically. One of the goals of the project is the alignment of visual and textual information. We will, for example, learn the correspondence between faces in an image and persons described in surrounding text. The role of the authors in the CLASS project is mainly on information extraction from text.

In the first phase of the project we build a classifier for automatic identification and categorization of entities in texts which we report here. This classifier extracts entities from text, and assigns a label to these entities chosen from an inventory of possible labels. This task is closely related to both named entity recognition (NER), which traditionally assigns nouns to a small number of categories and word sense disambiguation (Agirre and Rigau, 1996; Yarowsky, 1995), where the sense for a word is chosen from a much larger inventory of word senses.

We will employ a probabilistic model that's been used successfully in NER (Conditional Random Fields) and use this with an extensive inventory of word senses (the WordNet lexical database) to perform entity detection.

In section 2 we describe WordNet and it's use for entity categorization. Section 3 gives an overview of Conditional Random Fields and section 4 explains how the parameters of this model are estimated during training. We will drastically reduce the computational complexity of training in section 5. Section 6 describes the implementation of this method, section 7 the obtained results and finally section 8 future work.

2 WordNet

WordNet (Fellbaum et al., 1998) is a lexical database whose design is inspired by psycholinguistic theories of human lexical memory. English nouns, verbs, adjectives and adverbs are organized in synsets. A synset is a collection of words that have a close meaning and that represent an underlying concept. An example of such a synset is "person, individual, someone, somebody, mortal, soul". All these words refer to a human being.

WordNet (v2.1) contains 155.327 words, which are organized in 117.597 synsets. WordNet defines a number of relations between synsets. For nouns the most important relation is the hypernym/hyponym relation. A noun X is a hypernym of a noun Y if Y is a subtype or instance of X. For example, "bird" is a hypernym of "penguin" (and "penguin" is a hyponym of "bird"). This relation organizes the synsets in a hierarchical tree (Hayes, 1999), of which a fragment is pictured in fig. 1.

[1] http://class.inrialpes.fr/

Proceedings of the 2nd Workshop on Ontology Learning and Population, pages 33–40,
Sydney, July 2006. ©2006 Association for Computational Linguistics

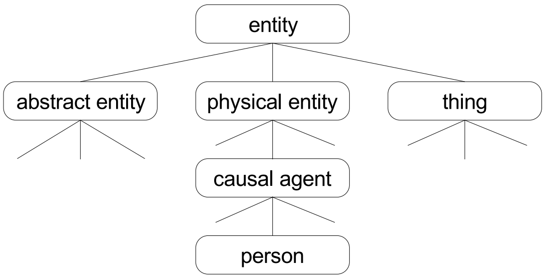


Figure 1: Fragment of the hypernym/hyponym tree

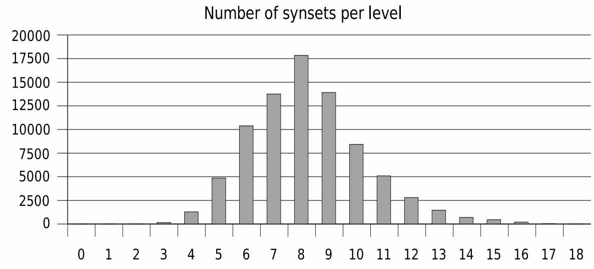


Figure 2: Number of synsets per level in WordNet

This tree has a depth of 18 levels and maximum width of 17837 synsets (fig. 2).

We will build a classifier using CRFs that tags noun phrases in a text with their WordNet synset. This will enable us to recognize entities, and to classify the entities in certain groups. Moreover, it allows learning the context pattern of a certain meaning of a word. Take for example the sentence "The ambulance took the remains of the bomber to the morgue." Having every noun phrase tagged with it's WordNet synset reveals that in this sentence, "bomber" is "a person who plants bombs" (and not "a military aircraft that drops bombs during flight"). Using the hypernym/hyponym relations from WordNet, we can also easily find out that "ambulance" is a kind of "car", which in turn is a kind of "conveyance, transport" which in turn is a "physical object".

3 Conditional Random Fields

Conditional random fields (CRFs) (Lafferty et al., 2001; Jordan, 1999; Wallach, 2004) is a statistical method based on undirected graphical models. Let X be a random variable over data sequences to be labeled and Y a random variable over corresponding label sequences. All components Y_i of Y are assumed to range over a finite label alphabet K. In this paper X will range over the sentences of a text, tagged with POS-labels and Y ranges over the synsets to be recognized in these sentences.

We define $G = (V, E)$ to be an undirected graph such that there is a node $v \in V$ corresponding to each of the random variables representing an element Y_v of Y. If each random variable Y_v obeys the Markov property with respect to G (e.g., in a first order model the transition probability depends only on the neighboring state), then the model (Y, X) is a Conditional Random Field. Although the structure of the graph G may be arbitrary, we limit the discussion here to graph structures in which the nodes corresponding to elements of Y form a simple first-order Markov chain.

A CRF defines a conditional probability distribution $p(Y|X)$ of label sequences given input sequences. We assume that the random variable sequences X and Y have the same length and use $\mathbf{x} = (x_1, ..., x_T)$ and $\mathbf{y} = (y_1, ..., y_T)$ for an input sequence and label sequence respectively. Instead of defining a joint distribution over both label and observation sequences, the model defines a conditional probability over labeled sequences. A novel observation sequence $\mathbf{x}$ is labeled with $\mathbf{y}$, so that the conditional probability $p(\mathbf{y}|\mathbf{x})$ is maximized. We define a set of K binary-valued features or feature functions $f_k(y_{t-1}, y_t, \mathbf{x})$ that each express some characteristic of the empirical distribution of the training data that should also hold in the model distribution. An example of such a feature is

$$f_k(y_{t-1}, y_t, \mathbf{x}) = \begin{cases} 1 & \begin{array}{l}\text{if } x \text{ has POS 'NN' and} \\ y_t \text{ is concept 'entity'}\end{array} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Feature functions can depend on the previous (y_{t-1}) and the current (y_t) state. Considering K feature functions, the conditional probability distribution defined by the CRF is

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} exp \left\{ \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(y_{t-1}, y_t, \mathbf{x}) \right\} \tag{2}$$

where λ_j is a parameter to model the observed statistics and $Z(\mathbf{x})$ is a normalizing constant computed as

$$Z(\mathbf{x}) = \sum_{\mathbf{y} \in Y} exp \left\{ \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(y_{t-1}, y_t, \mathbf{x}) \right\}$$

This method can be thought of a generalization of both the Maximum Entropy Markov model (MEMM) and the Hidden Markov model (HMM).

It brings together the best of discriminative models and generative models: (1) It can accommodate many statistically correlated features of the inputs, contrasting with generative models, which often require conditional independent assumptions in order to make the computations tractable and (2) it has the possibility of context-dependent learning by trading off decisions at different sequence positions to obtain a global optimal labeling. Because CRFs adhere to the maximum entropy principle, they offer a valid solution when learning from incomplete information. Given that in information extraction tasks, we often lack an annotated training set that covers all possible extraction patterns, this is a valuable asset.

Lafferty et al. (Lafferty et al., 2001) have shown that CRFs outperform both MEMM and HMM on synthetic data and on a part-of-speech tagging task. Furthermore, CRFs have been used successfully in information extraction (Peng and McCallum, 2004), named entity recognition (Li and McCallum, 2003; McCallum and Li, 2003) and sentence parsing (Sha and Pereira, 2003).

4 Parameter estimation

In this section we'll explain to some detail how to derive the parameters $\theta = \{\lambda_k\}$, given the training data. The problem can be considered as a constrained optimization problem, where we have to find a set of parameters which maximizes the log likelihood of the conditional distribution (McCallum, 2003). We are confronted with the problem of efficiently calculating the expectation of each feature function with respect to the CRF model distribution for every observation sequence x in the training data. Formally, we are given a set of training examples $D = \left\{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\right\}_{i=1}^{N}$ where each $\mathbf{x}^{(i)} = \left\{x_1^{(i)}, x_2^{(i)}, ..., x_T^{(i)}\right\}$ is a sequence of inputs and $\mathbf{y}^{(i)} = \left\{y_1^{(i)}, y_2^{(i)}, ..., y_T^{(i)}\right\}$ is a sequence of the desired labels. We will estimate the parameters by penalized maximum likelihood, optimizing the function:

$$l(\theta) = \sum_{i=1}^{N} log\, p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}) \tag{3}$$

After substituting the CRF model (2) in the likelihood (3), we get the following expression:

$$\begin{aligned} l(\theta) \quad &= \quad \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{k=1}^{K} \lambda_k f_k(y_{t-1}^{(i)}, y_t^{(i)}, \mathbf{x}^{(i)}) \\ &\quad\; - \sum_{i=1}^{N} log\, Z(\mathbf{x}^{(i)}) \end{aligned}$$

The function $l(\theta)$ cannot be maximized in closed form, so numerical optimization is used. The partial derivates are:

$$\begin{aligned} \frac{\partial l(\theta)}{\partial \lambda_k} \quad &= \quad \sum_{i=1}^{N}\sum_{t=1}^{T} f_k(y_t^{(i)}, y_{t-1}^{(i)}, \mathbf{x}^{(i)}) \\ &\quad\; - \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{y,y'} f_k(y', y, \mathbf{x}^{(i)})\, p(y', y|\mathbf{x}^{(i)}) \end{aligned} \tag{4}$$

Using these derivates, we can iteratively adjust the parameters θ (with Limited-Memory BFGS (Byrd et al., 1994)) until $l(\theta)$ has reached an optimum. During each iteration we have to calculate $p(y', y|x^{(i)})$. This can be done, as for the Hidden Markov Model, using the forward-backward algorithm (Baum and Petrie, 1966; Forney, 1996). This algorithm has a computational complexity of $O(TM^2)$ (where T is the length of the sequence and M the number of the labels). We have to execute the forward-backward algorithm once for every training instance during every iteration. The total cost of training a linear-chained CRFs is thus:

$$O(TM^2NG)$$

where N is the number of training examples and G the number of iterations. We've experienced that this complexity is an important delimiting factor when learning a big collection of labels. Employing CRFs to learn the 95076 WordNet synsets with 20133 training examples was not feasible on current hardware. In the next section we'll describe the method we've implemented to drastically reduce this complexity.

5 Reducing complexity

In this section we'll see how we create groups of features for every label that enable an important reduction in complexity of both labeling and training. We'll first discuss how these groups of features are created (section 5.1) and then how both labeling (section 5.2) and training (section 5.3) are performed using these groups.

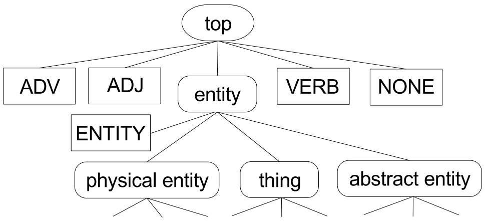


Figure 3: Fragment of the tree used for labeling

5.1 Hierarchical feature selection

To reduce the complexity of CRFs, we assign a selection of features to every node in the hierarchical tree. As discussed in section 2 WordNet defines a relation between synsets which organises the synsets in a tree. In its current form this tree does not meet our needs: we need a tree where every label used for labeling corresponds to exactly one leaf-node, and no label corresponds to a non-leaf node. We therefor modify the existing tree. We create a new top node (“top”) and add the original tree as defined by WordNet as a subtree to this top-node. We add leaf-nodes corresponding to the labels “NONE”, “ADJ”, “ADV”, “VERB” to the top-node and for the other labels (the noun synsets) we add a leaf-node to the node representing the corresponding synset. For example, we add a node corresponding to the label “ENTITY” to the node “entity”. Fig. 3 pictures a fraction of this tree. Nodes corresponding to a label have an uppercase name, nodes not corresponding to a label have a lowercase name.

We use v to denote nodes of the tree. We call the top concept v^{top} and the concept v^+ the parent of v, which is the parent of v^-. We call A_v the collection of ancestors of a concept v, including v itself.

We will now show how we transform a regular CRF in a CRF that uses hierarchical feature selection. We first notice that we can rewrite eq. 2 as

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} G(y_{t-1}, y_t, \mathbf{x})$$

with $G(y_{t-1}, y_t, \mathbf{x}) = exp(\sum_{k=1}^{K} \lambda_k f_k(y_{t-1}, y_t, \mathbf{x}))$

We rewrite this equation because it will enable us to reduce the complexity of CRFs and it has the property that $p(y_t|y_{t-1}, \mathbf{x}) \approx G(y_{t-1}, y_t, \mathbf{x})$ which we will use in section 5.3.

We now define a collection of features F_v for every node v. If v is leaf-node, we define F_v as the collection of features $f_k(y_{t-1}, y_t, \mathbf{x})$ for which it is possible to find a node v_{t-1} and input $\mathbf{x}$ for which $f_k(v_{t-1}, v, \mathbf{x}) \neq 0$. If v is a non-leaf node, we define F_v as the collection of features $f_k(y_{t-1}, y_t, \mathbf{x})$ (1) which are elements of F_{v^-} for every child node v^- of v and (2) for every v_1^- and v_2^-, children of v, it is valid that for every previous label v_{t-1} and input $\mathbf{x}$ $f_k(v_{t-1}, v_1^-, \mathbf{x}) = f_k(v_{t-1}, v_2^-, \mathbf{x})$.

Informally, F_v is the collection of features which are useful to evaluate for a certain node. For the leaf-nodes, this is the collection of features that can possibly return a non-zero value. For non-leaf nodes, it’s useful to evaluate features belonging to F_v when they have the same value for all the descendants of that node (which we can put to good use, see further).

We define $F'_v = F_v \backslash F_{v^+}$ where v^+ is the parent of label v. For the top node v^{top} we define $F'_{v^{top}} = F_{v^{top}}$. We also set

$$G'(y_{t-1}, y_t, \mathbf{x}) = exp \left\{ \sum_{f_k \in F'_{y_t}} \lambda_k f_k(y_{t-1}, y_t, \mathbf{x}) \right\}$$

We’ve now organised the collection of features in such a way that we can use the hierarchical relations defined by WordNet when determining the probability of a certain labeling $\mathbf{y}$. We first see that

$$\begin{aligned} G(y_{t-1}, y_t, \mathbf{x}) &= exp \left\{ \sum_{f_k \in F_{y_t}} \lambda_k f_k(y_{t-1}, y_t, \mathbf{x}) \right\} \\ &= G(y_{t-1}, y_t^+, x) G'(y_{t-1}, y_t, x) \\ &= \ldots \\ &= \prod_{v \in A_{y_t}} G'(y_{t-1}, v, x) \end{aligned}$$

we can now determine the probability of a labeling $\mathbf{y}$, given input $\mathbf{x}$

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} \prod_{v \in A_{y_t}} G'(y_{t-1}, v, \mathbf{x}) \qquad (5)$$

This formula has exactly the same result as eq. 2. Because we assigned a collection of features to every node, we can discard parts of the search space when searching for possible labelings, obtaining an important reduction in complexity. We elaborate this idea in the following sections for both labeling and training.

5.2 Labeling

The standard method to label a sentence with CRFs is by using the Viterbi algorithm (Forney, 1973; Viterbi, 1967) which has a computational complexity of $O(TM^2)$. The basic idea to reduce this computational complexity is to select the best labeling in a number of iterations. In the first iteration, we label every word in a sentence with a label chosen from the top-level labels. After choosing the best labeling, we refine our choice (choose a child label of the previous chosen label) in subsequent iterations until we arrive at a synset which has no children. In every iteration we only have to choose from a very small number of labels, thus breaking down the problem of selecting the correct label from a large number of labels in a number of smaller problems.

Formally, when labeling a sentence we find the label sequence $\mathbf{y}$ such that $\mathbf{y}$ has the maximum probability of all labelings. We will estimate the best labeling in an iterative way: we start with the best labeling $\mathbf{y}^{top-1} = \{y_1^{top-1}, ..., y_T^{top-1}\}$ choosing only from the children $y_t^{top-1}$of the top node. The probability of this labeling $\mathbf{y}^{top-1}$ is

$$p(\mathbf{y}^{top-1}|\mathbf{x}) = \frac{1}{Z'(\mathbf{x})} \prod_{t=1}^{T} G'(y_{t-1}, y_t^{top-1}, \mathbf{x})$$

where $Z'(x)$ is an appropriate normalizing constant. We now select a labeling $\mathbf{y}^{top-2}$ so that on every position t node y_t^{top-2} is a child of y_t^{top-1}. The probabilty of this labeling is (following eq. 5)

$$p(\mathbf{y}^{top-2}|\mathbf{x}) = \frac{1}{Z'(\mathbf{x})} \prod_{t=1}^{T} \prod_{v \in A_{y_t^{top-2}}} G'(y_{t-1}, v, \mathbf{x})$$

After selecting a labeling $\mathbf{y}^{top-2}$ with maximum probability, we proceed by selecting a labeling $\mathbf{y}^{top-3}$ with maximum probability etc.. We proceed using this method until we reach a labeling in which every y_t is a node which has no children and return this labeling as the final labeling.

The assumption we make here is that if a node v is selected at position t of the most probable labeling $\mathbf{y}^{top-s}$ the children v^- have a larger probability of being selected at position t in the most probable labeling $\mathbf{y}^{top-s-1}$. We reduce the number of labels we take into consideration by stating that for every concept v for which $v \neq y_t^{top-s}$, we set $G'(y_{t-1}, v_t^-, \mathbf{x}) = 0$ for every child v^- of v. This reduces the space of possible labelings drastically, reducing the computational complexity of the Viterbi algorithm. If q is the average number of children of a concept, the depth of the tree is $log_q(M)$. On every level we have to execute the Viterbi algorithm for q labels, thus resulting in a total complexity of

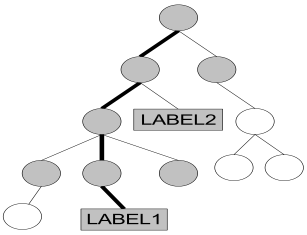


Figure 4: Nodes that need to be taken into account during the forward-backward algorithm

$$O(T\, log_q(M) q^2) \tag{6}$$

5.3 Training

We will now discuss how we reduce the computational complexity of training. As explained in section 4 we have to estimate the parameters λ_k that optimize the function $l(\theta)$. We will show here how we can reduce the computational complexity of the calculation of the partial derivates $\frac{\partial l(\theta)}{\partial \lambda_k}$ (eq. 4). The predominant factor with regard to the computational complexity in the evaluation of this equation is the calculation of $p(y_{t-1}, y|\mathbf{x}^{(i)})$. Recall we do this with the forward-backward algorithm, which has a computational complexity of $O(TM^2)$. We reduce the number of labels to improve performance. We will do this by making the same assumption as in the previous section: for every concept v at level s, for which $v \neq y_t^{top-s}$, we set $G'(y_{t-1}, v_t^-, \mathbf{x}) = 0$ for every child v^- of v. Since (as noted in sect. 5.2) $p(v_t|y_{t-1}, \mathbf{x}) \approx G(y_{t-1}, v_t, \mathbf{x})$, this has the consequence that $p(v_t|y_{t-1}, \mathbf{x}) = 0$ and that $p(v_t, y_{t-1}|\mathbf{x}) = 0$. Fig. 4 gives a graphical representation of this reduction of the search space. The correct label here is “LABEL1” , the grey nodes have a non-zero $p(v_t, y_{t-1}|\mathbf{x})$ and the white nodes have a zero $p(v_t, y_{t-1}|\mathbf{x})$.

In the forward backward algorithm we only have to account every node v that has a non-zero $p(v, y_{t-1}|\mathbf{x})$. As can be easily seen from fig. 4, the number of nodes is $qlog_qM$, where q is the average number of children of a concept. The total complexity of running the forward-backward algorithm is $O(T(q\, log_qM)^2)$. Since we have to run this algorithm once for every gradient compu-

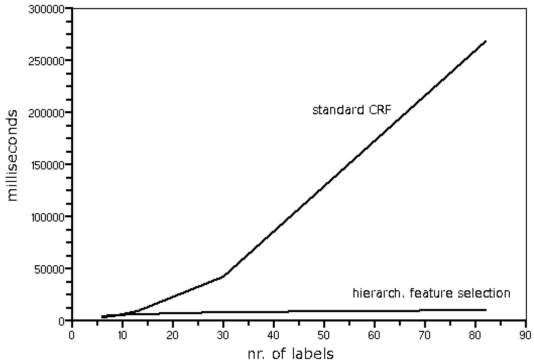


Figure 5: Time needed for one training cycle

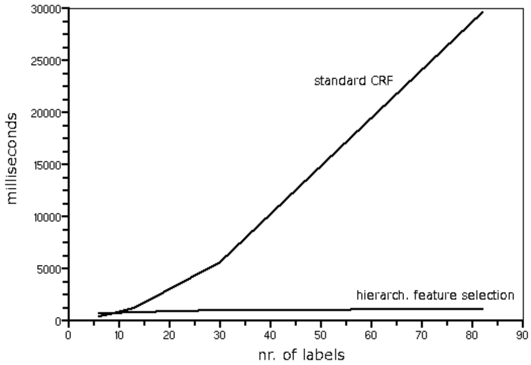


Figure 6: Time needed for labeling

tation for every training instance we find the total training cost

$$O(T(q\, log_q M)^2 NG) \tag{7}$$

6 Implementation

To implement the described method we need two components: an interface to the WordNet database and an implementation of CRFs using a hierarchical model. JWordNet is a Java interface to WordNet developed by Oliver Steele (which can be found on `http://jwn.sourceforge.net/`). We used this interface to extract the WordNet hierarchy.

An implementation of CRFs using the hierarchical model was obtained by adapting the Mallet[2] package. The Mallet package (McCallum, 2002) is an integrated collection of Java code useful for statistical natural language processing, document classification, clustering, and information extraction. It also offers an efficient implementation of CRFs. We've adapted this implementation so it creates hierarchical selections of features which are then used for training and labeling.

We used the Semcor corpus (Fellbaum et al., 1998; Landes et al., 1998) for training. This corpus, which was created by the Princeton University, is a subset of the English Brown corpus containing almost 700,000 words. Every sentence in the corpus is noun phrase chunked. The chunks are tagged by POS and both noun and verb phrases are tagged with their WordNet sense. Since we do not want to learn a classification for verb synsets, we replace the tags of the verbs with one tag "VERB".

7 Results

The major goal of this paper was to build a classifier that could learn all the WordNet synsets in a reasonable amount of time. We will first discuss the improvement in time needed for training and labeling and then discuss accuracy.

We want to test the influence of the number of labels on the time needed for training. Therefor, we created different training sets, all of which had the same input (246 sentences tagged with POS labels), but a different number of labels. The first training set only had 5 labels ("ADJ", "ADV", "VERB", "entity" and "NONE"). The second had the same labels except we replaced the label "entity" with either "physical entity", "abstract entity" or "thing". We continued this procedure, replacing parent nouns labels with their children (i.e. hyponyms) for subsequent training sets. We then trained both a CRF using a hierarchical feature selection and a standard CRF on these training sets.

Fig. 5 shows the time needed for one iteration of training with different numbers of labels. We can see how the time needed for training slowly increases for the CRF using hierarchical feature selection but increases fast when using a standard CRF. This is conform to eq. 7.

Fig. 6 shows the average time needed for labeling a sentence. Here again the time increases slowly for a CRF using hierarchical feature selection, but increases fast for a standard CRF, conform to eq. 6.

Finally, fig 7 shows the error rate (on the training data) after each training cycle. We see that a standard CRF and a CRF using hierarchical feature selection perform comparable. Note that fig 7 gives the error rate on the training data but this

[2] http://mallet.cs.umass.edu/

can differ considerable from the error rate on unseen data.

After these tests on a small section of the Semcor corpus, we trained a CRF using hierarchical feature selection on 7/8 of the full corpus. We trained for 23 iterations, which took approximately 102 hours. Testing the model on the remaining 1/8 of the corpus resulted in an accuracy of 77.82%. As reported in (McCarthy et al., 2004), a baseline approach that ignors context but simply assigns the most likely sense to a given word obtains a accuracy of 67%. We did not have the possibility to compare the accuracy of this model with a standard CRF, since as already stated, training such a CRF takes impractically long, but we can compare our systems with existing WSD-systems. Mihalcea and Moldovan (Mihalcea and Moldovan, 1999) use the semantic density between words to determine the word sense. They achieve an accuracy of 86.5% (testing on the first two tagged files of the Semcor corpus). Wilks and Stevenson (Wilks and Stevenson, 1998) use a combination of knowledge sources and achieve an accuracy of 92%[3]. Note that both these methods use additional knowledge apart from the WordNet hierarchy.

The sentences in the training and testing sets were already (perfectly) POS-tagged and noun chunked, and that in a real-life situation additional preprocessing by a POS-tagger (such as the LT-POS-tagger[4]) and noun chunker (such as described in (Ramshaw and Marcus, 1995)) which will introduce additional errors.

8 Future work

In this section we'll discuss some of the work we plan to do in the future. First of all we wish to evaluate our algorithm on standard test sets, such as the data of the Senseval conference[5], which tests performance on word sense disambiguation, and the data of the CoNLL 2003 shared task[6], on named entity recognition.

An important weakness of our algorithm is the fact that, to label a sentence, we have to traverse the hierarchy tree and choose the correct synsets at every level. An error at a certain level can not be recovered. Therefor, we would like to perform some a of beam-search (Bisiani, 1992), keeping a number of best labelings at every level. We strongly suspect this will have a positive impact on the accuracy of our algorithm.

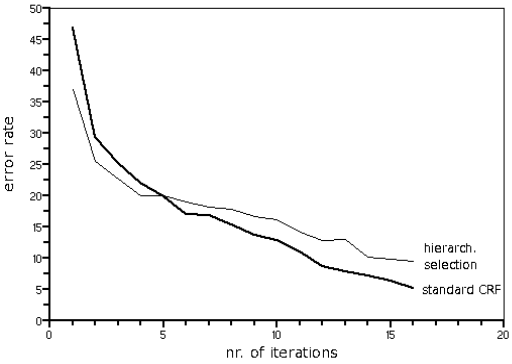


Figure 7: Error rate during training

As already mentioned, this work is carried out during the CLASS project. In the second phase of this project we will discover classes and attributes of entities in texts. To accomplish this we will not only need to label nouns with their synset, but we also need to label verbs, adjectives and adverbs. This can become problematic as WordNet has no hypernym/hyponym relation (or equivalent) for the synsets of adjectives and adverbs. WordNet has an equivalent relation for verbs (hypernym/troponym), but this structures the verb synsets in a big number of loosely structured trees, which is less suitable for the described method. VerbNet (Kipper et al., 2000) seems a more promising resource to use when classifying verbs, and we will also investigate the use of other lexical databases, such as ThoughtTreasure (Mueller, 1998), Cyc (Lenat, 1995), Openmind Commonsense (Stork, 1999) and FrameNet (Baker et al., 1998).

Acknowledgments

The work reported in this paper was supported by the EU-IST project CLASS (Cognitive-Level Annotation using Latent Statistical Structure, IST-027978).

References

Eneko Agirre and German Rigau. 1996. Word sense disambiguation using conceptual density. In *Proceedings of the 16th International Conference on*

[3] This method was tested on the Semcore corpus, but use the word senses of the Longman Dictionary of Contemporary English

[4] http://www.ltg.ed.ac.uk/software/

[5] http://www.senseval.org/

[6] http://www.cnts.ua.ac.be/conll2003/

Computational Linguistics (Coling'96), pages 16–22, Copenhagen, Denmark.

C. F. Baker, C. J. Fillmore, and J. B. Lowe. 1998. The Berkeley Framenet project. In *Proceedings of the COLING-ACL*.

L. E. Baum and T. Petrie. 1966. Statistical inference for probabilistic functions of finite state markov chains. *Annals of Mathematical Statistics,*, 37:1554–1563.

R. Bisiani. 1992. Beam search. In S. C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, New York. Wiley-Interscience.

Richard H. Byrd, Jorge Nocedal, and Robert B. Schnabel. 1994. Representations of quasi-newton matrices and their use in limited memory methods. *Math. Program.*, 63(2):129–156.

C. Fellbaum, J. Grabowski, and S. Landes. 1998. Performance and confidence in a semantic annotation task. In C. Fellbaum, editor, *WordNet: An Electronic Lexical Database*. The MIT Press.

G. D. Forney. 1973. The viterbi algorithm. In *Proceeding of the IEEE*, pages 268 – 278.

G. D. Forney. 1996. The forward-backward algorithm. In *Proceedings of the 34th Allerton Conference on Communications, Control and Computing*, pages 432–446.

Brian Hayes. 1999. The web of words. *American Scientist*, 87(2):108–112, March-April.

Michael I. Jordan, editor. 1999. *Learning in Graphical Models.* The MIT Press, Cambridge.

K. Kipper, H.T. Dang, and M. Palmer. 2000. Class-based construction of a verb lexicon. *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-2000)*.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*.

S. Landes, C. Leacock, and R.I. Tengi. 1998. Building semantic concordances. In C. Fellbaum, editor, *WordNet: An Electronic Lexical Database*. The MIT Press.

D. B. Lenat. 1995. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):32–38.

Wei Li and Andrew McCallum. 2003. Rapid development of hindi named entity recognition using conditional random fields and feature induction. *ACM Transactions on Asian Language Information Processing (TALIP)*, 2(3):290–294.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 188–191. Edmonton, Canada.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

A. McCallum. 2003. Efficiently inducing features of conditional random fields. In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*.

D. McCarthy, R. Koeling, J. Weeds, and J. Carroll. 2004. Using automatically acquired predominant senses for word sense disambiguation. In *Proceedings of the ACL SENSEVAL-3 workshop*, pages 151–154, Barcelona, Spain.

R. Mihalcea and D.I. Moldovan. 1999. A method for word sense disambiguation of unrestricted text. In *Proceedings of the 37th conference on Association for Computational Linguistics*, pages 152–158. Association for Computational Linguistics Morristown, NJ, USA.

Erik T. Mueller. 1998. *Natural language processing with ThoughtTreasure*. Signiform, New York.

F. Peng and A. McCallum. 2004. Accurate information extraction from research papers using conditional random fields. In *Proceedings of Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 329–336.

L.A. Ramshaw and M.P. Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of the Third ACL Workshop on Very Large Corpora*, pages 82–94. Cambridge MA, USA.

F. Sha and F. Pereira. 2003. Shallow parsing with conditional random fields. In *Proceedings of Human Language Technology, HLT-NAACL*.

D. Stork. 1999. The openmind initiative. *IEEE Intelligent Systems & their applications*, 14(3):19–20.

A. J. Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. Informat. Theory*, 13:260–269.

Hanna M. Wallach. 2004. Conditional random fields: An introduction. Technical Report MS-CIS-04-21., University of Pennsylvania CIS.

Y. Wilks and M. Stevenson. 1998. Word sense disambiguation using optimised combinations of knowledge sources. *Proceedings of COLING/ACL*, 98.

D. Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196.

Taxonomy Learning using Term Specificity and Similarity

Pum-Mo Ryu

Computer Science Division, KAIST
KORTERM/BOLA
Korea
pmryu@world.kaist.ac.kr

Key-Sun Choi

Computer Science Division, KAIST
KORTERM/BOLA
Korea
kschoi@cs.kaist.ac.kr

Abstract

Learning taxonomy for technical terms is difficult and tedious task, especially when new terms should be included. The goal of this paper is to assign taxonomic relations among technical terms. We propose new approach to the problem that relies on term specificity and similarity measures. Term specificity and similarity are necessary conditions for taxonomy learning, because highly specific terms tend to locate in deep levels and semantically similar terms are close to each other in taxonomy. We analyzed various features used in previous researches in view of term specificity and similarity, and applied optimal features for term specificity and similarity to our method.

1 Introduction

Taxonomy is a collection of controlled vocabulary terms organized into a hierarchical structure. Each term in a taxonomy is one or more parent-child relationships to other terms in the taxonomy. Taxonomies are useful artifacts for organizing many aspects of knowledge. As components of ontologies, taxonomies can provide an organizational model for a domain (domain ontology), or a model suitable for specific tasks (task ontologies) (Burgun & Bodenreider, 2001). However their wide usage is still hindered by time-consuming, cost-ineffective building processes.

The main paradigms of taxonomy learning are on the one hand pattern based approaches and on the other hand distributional hypothesis based approaches. The former is approaches based on matching lexico-syntactic patterns which convey taxonomic relations in a corpus (Hearst, 1992; Iwanska et al., 2000), and the latter is statistical approaches based on the distribution of context in corpus (Cimiano et al., 2005; Yamamoto et al., 2005; Sanderson & Croft, 1999). The former features a high precision and low recall compared to the latter. The quality of learned relations is higher than those of statistical approaches, while the patterns are rarely applied in real corpus. It is also difficult to improve performance of pattern based approaches because they are simple and clear. So, many researches have been focused on raising precision of statistical approaches.

We introduce new distributional hypothesis based taxonomy learning method using term specificity and term similarity. Term specificity is a measure of information quantity of terms in given domain. When a term has much domain information, the term is highly specific to the domain, and vice versa (Ryu & Choi, 2005). Because highly specific terms tend to locate in low level in domain taxonomy, term specificity can be used as a necessary condition for taxonomy learning. Term similarity is degree of semantic overlap among terms. When two terms share many common characteristics, they are semantically similar to each other. Term similarity can be another necessary condition for taxonomy learning, because semantically similar terms locate near by in given domain taxonomy. The two conditions are generally valid for terms in a taxonomic relation, while terms satisfying the conditions do not always have taxonomic relation. So they are necessary conditions for taxonomy learning.

Based on these conditions, it is highly probable that term t_1 is an ancestor of term t_2 in domain taxonomy T_D, when t_1 and t_2 are semantically similar enough and the specificity of t_1 is lower than that of t_2 in D as in Figure 1. However, t_1 is not an ancestor of t_3 even though the speci-

Proceedings of the 2nd Workshop on Ontology Learning and Population, pages 41–48,
Sydney, July 2006. ©2006 Association for Computational Linguistics

ficity of t_1 is lower than that of t_3 because t_1 is not similar to t_3 on the semantic level.

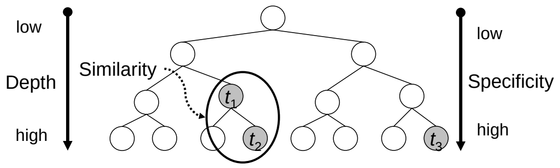


Figure 1. Term specificity and term similarity in a domain taxonomy T_D

The strength of this method lies in its ability to adopt different optimal features for term specificity and term similarity. Most of current researches relied on single feature such as adjectives of terms, verb-argument relation, or co-occurrence ratio in documents according to their methods. Firstly, we analyze characteristics of features for taxonomy learning in view of term specificity and term similarity to show that the features embed characteristics of specificity and similarity, and finally apply optimal features to our method.

Additionally we tested inside information of terms to measure term specificity and similarity. As multiword terms cover the larger part of technical terms, lexical components are featuring information representing semantics of terms (Cerbah, 2000).

The remainder of this paper is organized follows. Characteristics of term specificity are described in Section 2, while term similarity and its features are addressed in Section 3. Our taxonomy learning method is discussed in Section 4. Experiment and evaluation are discussed in Section 5, and finally, conclusions are drawn in Section 6.

2 Term Specificity

Specificity is degree of detailed information of an object about given target object. For example, if an encyclopedia contains detailed information about '*IT domain*', then the encyclopedia is '*IT specific encyclopedia*'. In this context, specificity is a function of objects and target object to real number. Traditionally term specificity is widely used in information retrieval systems to weight index terms in documents (S. Jones, 1972; Aizawa, 2003; Wong & Yao, 1992). In information retrieval context, term specificity is function of index terms and documents. On the other hand, term specificity is the function of terms and target domains in taxonomy learning context (Ryu & Choi 2005). Term specificity to a domain is quantified to a positive real number as shown in Eq. (1).

$$Spec(t \mid D) \in R^{+} \tag{1}$$

where t is a term, and $Spec(t|D)$ is the specificity of t in a given domain D. We simply use $Spec(t)$ instead of $Spec(t|D)$ assuming a particular domain D in this paper.

Understanding the relation between domain concepts and their lexicalization methods is needed, before we describe term specificity measuring methods. Domain specific concepts can be distinguished by a set of what we call '*characteristics*'. More specific concepts are created by adding characteristics to the set of characteristics of existing concepts. Let us consider two concepts: C_1 and C_2. C_1 is an existing concept and C_2 is a newly created concept by combining new characteristics to the characteristic set of C_1. In this case, C_1 is an ancestor of C_2 (ISO, 2000). When domain specific concepts are lexicalized as terms, the terms' word-formation is classified into two categories based on the composition of component words. In the first category, new terms are created by adding modifiers to existing terms. Figure 2 shows a subtree of financial ontology. For example '*current asset*' was created by adding the modifier '*current*' to its hypernym '*asset*'. In this case, inside information is a good evidence to represent the characteristics. In the second category, new terms are created independently of existing terms. For example, '*cache*', '*inventory*', and '*receivable*' share no common words with their hypernyms '*current asset*' and '*asset*'. In this case, outside information is used to differentiate the characteristics of the terms.

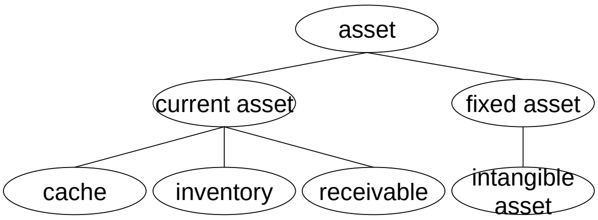


Figure 2. Subtree of financial ontology

There are many kinds of inside and outside information to be used in measuring term specificity. Distribution of adjective-term relation and verb-argument dependency relation are collocation based statistics. Distribution of adjective-term relation refers to the idea that specific nouns are rarely modified, while general nouns are frequently modified in text. This feature has been discussed to measure specificity of nouns in (Caraballo, 1999; Ryu & Choi, 2005) and to build taxonomy of Japanese nouns (Yamamoto et al., 2005). Inversed specificity of a term can be measured by entropy of adjectives as shown Eq. (2).

$$Spec_{adj}(t)^{-1} = -\sum_{adj} P(adj \mid t)\log P(adj \mid t) \qquad (2)$$

where $P(adj|t)$, the probability that *adj* modifies *t*, is estimated as *freq*(*adj*,*t*)/*freq*(*t*). The entropy is the average information quantity of all (*adj*,*t*) pairs for term *t*. Specific terms have low entropy, because their adjective distributions are simple.

For verb-argument distribution, we assume that domain specific terms co-occur with selected verbs which represent special characteristics of terms while general terms are associated with multiple verbs. Under this assumption, we make use of syntactic dependencies between verbs appearing in the corpus and their arguments such as subjects and objects. For example, '*inventory*'[1], in Figure 2, shows a tendency to be objects of specific verbs like '*increase*' and '*reduce*'. This feature was used in (Cimiano et al., 2005) to learn concept hierarchy. Inversed specificity of a term can be measured by entropy of verb-argument relations as Eq. (3).

$$Spec_{v_{arg}}(t)^{-1} = -\sum_{v_{arg}} P(t \mid v_{arg})\log P(t \mid v_{arg}) \qquad (3)$$

where $P(t|v_{arg})$, the probability that *t* is argument of v_{arg}, is estimated as $freq(t,v_{arg})/freq(v_{arg})$. The entropy is the average information quantity of all (t,v_{arg}) pairs for term *t*.

Conditional probability of term co-occurrence in documents was used in (Sanderson & Croft, 1999) to build term taxonomy. This statistics is based on the assumption that, for two terms, t_i and t_j, t_i is said to subsume t_j if the following two conditions hold,

$$P(t_i|t_j) = 1 \text{ and } P(t_j|t_i)<1 \qquad (4)$$

In other words, t_i subsumes t_j if the documents which t_j occurs in are a subset of the documents which t_i occurs in, therefore t_i can be parent of t_j in taxonomy. Although a good number of term pairs are found that adhere to the two subsumption conditions, it is noticed that many are just failing to be included because a few occurrences of the subsumed term, t_j, does not co-occur with t_i. Subsequently, the conditions are relaxed and *subsume* function is defined as Eq. (5). In case of $P(t_i|t_j)>P(t_j|t_i)$, $subsume(t_i,t_j)$ returns 1, otherwise returns 0.

$$subsume(t_i,t_j) = \begin{cases} 1 & \text{if } P(t_i \mid t_j) > P(t_j \mid t_i) \\ 0 & \text{otherwise} \end{cases} \qquad (5)$$

We apply this function to calculate term specificity as shown Eq. (6) where a term is specific when it is subsumed by most of other terms. Specificity of *t* is determined by the ratio of terms that subsume *t* over all co-occurring terms.

$$Spec_{coldoc}(t) = \frac{\sum_{1\le j\le n} subsume(t_j,t)}{n} \qquad (6)$$

where *n* is number of terms co-occurring terms with *t*.

Finally, inside-word information is important to compute specificity for multiword terms. Consider a term *t* that consists of two words like $t = w_1w_2$. Two words, w_1 and w_2, have their unique characteristics and the characteristics are summed up to the characteristic of *t*. Mutual information is used to estimate the association between a term and its component words. Let $T=\{t_1,\ldots,t_N\}$ be a set of terms found in a corpus, and $W=\{w_1,\ldots,w_M\}$ be a set of component words composing the terms in *T*. Assume a joint probability distribution $P(t_i,w_j)$, probability of w_j is a component of t_i, is given for t_i and w_j. Mutual information between t_i and w_j compares the probability of observing t_i and w_j together and the probability of observing t_i and w_j independently. The mutual information represents the reduction of uncertainty about t_i when w_j is observed. The summed mutual information between t_i and *W*, as in Eq. (7), is total reduction of uncertainty about t_i when all component words are observed.

$$Spec_{in}(t_i) = \sum_{w_j\in W} \log\frac{P(t_i,w_j)}{P(t_i)P(w_j)} \qquad (7)$$

This equation indicates that w_j which is highly associated to t_i contributes specificity of t_i. For example, '*debenture bond*' is more specific concept than '*financial product*'. Intuitively, '*debenture*' is highly associated to '*debenture bond*'

[1] '*Inventory*' consists of a list of goods and materials held available in stock (http://en.wikipedia.org/wiki/Inventory).

compared with '*bond*' to '*debenture bond*' or '*financial*', '*product*' to '*financial product*'.

3 Term Similarity

We evaluate four statistical and lexical features, related to taxonomy learning, in view of term similarity. Three statistical features have been used in existing taxonomy learning researches.

(Sanderson & Croft, 1999) used conditional probability of co-occurring terms in same document in taxonomy learning process as shown in Eq. (4). This feature can be used to measure similarity of terms. If two terms co-occur in common documents, they are semantically similar to each other. Based on this assumption, we can calculate term similarity by comparing the frequency of co-occurring t_i and t_j together and the frequency of occurring t_i and t_j independently, as Eq. (8).

$$Sim_{coldoc}(t_i, t_j) = \frac{2 * df(t_i, t_j)}{df(t_i) + df(t_j)} \qquad (8)$$

where $df(t_i,t_j)$ is number of documents in which both t_i and t_j co-occur, $df(t_i)$ is number of documents in which t_i occurs.

(Yamamoto et al., 2005) used adjective patterns to make characteristics vectors for terms in Complementary Similarity Measure (CSM). Although CSM was initially designed to extract superordinate-subordinate relations, it is a similarity measure by itself. They proposed two CSM measures; one is for binary images in which values in feature vectors are 0 or 1, and the other is for gray-scale images in which values in feature vectors are 0 through 1. We adapt gray-scale measure in similarity calculation, because it showed better performance in their research.

(Cimiano et al., 2005) applied Formal Concept Analysis (FCA) to extract taxonomies from a text corpus. They modeled the context of a term as a vector representing syntactic dependencies. Similarity based on verb-argument dependencies is calculated using cosine measure as Eq. (9).

$$Sim_{v_{arg}}(t_i, t_j) = \frac{\sum_{v_{arg} \in V} P(t_i \mid v_{arg}) P(t_j \mid v_{arg})}{\sqrt{\sum_{v_{arg} \in V} P(t_i \mid v_{arg})^2} \sqrt{\sum_{v_{arg} \in V} P(t_j \mid v_{arg})^2}} \qquad (9)$$

where $P(t|v_{arg})$, the probability that t is argument of v_{arg}, is estimated as $freq(t,v_{arg})/freq(v_{arg})$. Above three similarity measures are valid when terms, t_i and t_j, appear in corpus one or more times.

The last similarity measure is based on inside information of terms. Because many domain terms are multiword terms, component words are clues for term similarity. If two terms share many common words, they share common characteristics in given domain. For example, four words '*asset*', '*current asset*', '*fixed asset*' and '*intangible asset*' share characteristics related to '*asset*' as in Figure 2. This similarity measure is shown in Eq. (10).

$$Sim_{in}(t_i, t_j) = \frac{2 * cwc(t_i, t_j)}{|t_i| + |t_j|} \qquad (10)$$

where $|t|$ is word count of t, and $cwc(t_i,t_j)$ is common word count in t_i and t_j. $Sim_{in}(t_i,t_j)$ is valid when $cwc(t_i,t_j)>0$. Because $cwc(t_i,t_j)=0$ for most of term pairs, it is difficult to catch reliable results for all possible term pairs.

4 Taxonomy Learning Process

We model taxonomy learning process as a sequential insertion of new terms to current taxonomy. New taxonomy starts with empty state, and changes to rich taxonomic structure with the repeated insertion of terms as depicted in Figure 3. Terms to be inserted are sorted by term specificity values. Term insertion based on the increasing order of term specificity is natural, because the taxonomy grows from top to down with term insertion process in increasing specificity sequence.

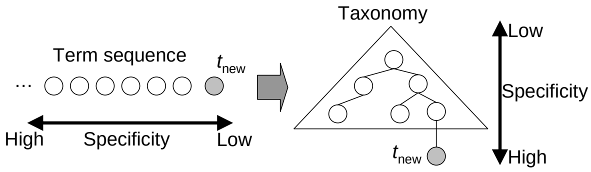


Figure 3. Terms are inserted to taxonomy in the sequence of specificity

According to above assumption, our system selects possible hypernyms of a new term, t_{new} in current taxonomy as following steps:

- Step 1: Select n-most similar terms to t_{new} from current taxonomy
- Step 2: Select candidate hypernyms of t_{new} from n-most similar terms. Specificity of candidate hypernyms is less than that of t_{new}.

- Step 3: Insert t_{new} as hyponyms of candidate hypernyms

For example, suppose t_2, t_4, t_5 and t_6, are four most similar terms to t_{new} in Figure 4. Two terms t_2 and t_4 are selected as candidate hypernyms of t_{new}, because specificity of the terms is less than specificity of t_{new}.

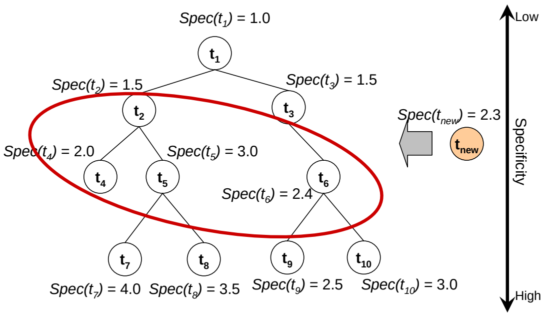


Figure 4. Selection of candidate hypernyms of t_{new} from taxonomy using term specificity and similarity

5 Experiment and Evaluation

We applied our taxonomy learning method to set of terms in existing taxonomy. We removed all relations from the taxonomy, and made new taxonomic relations among the terms. The learned taxonomy was then compared to original taxonomy. Our experiment is composed of four steps. Firstly, we calculated term specificity using specificity measures discussed in chapter 2, secondly, we calculated term similarity using similarity measures described in chapter 3, thirdly, we applied the best specificity and similarity features to our taxonomy building process, and finally, we evaluated our method and compared with other taxonomy learning methods.

Finance ontology [2] which was developed within the GETESS project (Staab et al., 1999) was used in our experiment. We slightly modified original ontology. We unified different expressions of same concept to identical expression. For example, '*cd-rom drive*' and '*cdrom drive*' are unified as '*cd-rom drive*' because the former is more usual expression than the latter. We also removed terms that are not descendents of '*root*' node to make the taxonomy have single root node. The taxonomy consists of total 1,819 nodes and 1,130 distinct nodes. Maximum and average depths are 15 and 5.5 respectively, and maximum and average children nodes are 32 and 3.5 respectively.

We considered Reuters21578[3] corpus, over 3.1 million words in title and body fields. We parsed the corpus using Connexor functional dependency parser[4] and extracted various statistics: term frequency, distribution of adjectives, distribution of co-occurring frequency in documents, and verb-argument distribution.

5.1 Term Specificity

Term specificity was evaluated based on three criteria: recall, precision and F-measure. Recall is the fraction of the terms that have specificity values by the given measuring method. Precision is the fraction of relations with correct specificity values. F-measure is a harmonic mean of precision and recall into a single measure of overall performance. Precision (P_{spec}), recall (R_{spec}), F-measure (F_{spec}) is defined as follows:

$$R_{spec} = \frac{\#\ of\ terms\ with\ specificity}{\#\ of\ all\ terms}$$
$$P_{spec} = \frac{\#\ of\ R_{valid}(p,c)\ with\ correct\ specificity}{\#\ of\ R_{valid}(p,c)} \tag{11}$$

where $R_{valid}(p,c)$ is a valid parent-child relation in original taxonomy, and a relation is *valid* when the specificity of two terms are measured by the given method. If the specificity of child term, *c*, is larger than that of parent term, *p*, then the relation is *correct*.

We tested four specificity measuring methods discussed in section 2 and the result is shown in Table 1. $Spec_{\text{adj}}$ showed the highest precision as we anticipated. Because domain specific terms have sufficient information in themselves; they are rarely modified by other words in real text. However, $Spec_{\text{adj}}$ showed the lowest recall for data sparseness problem. As mentioned above, it is hard to collect sufficient adjectives for domain specific terms from text. $Spec_{\text{varg}}$ showed the lowest precision. This result indicates that distribution of verb-argument relation is less correlated to term specificity. $Spec_{\text{in}}$ showed the highest recall because it measures term specificity using component words contrary to other methods. $Spec_{\text{coldoc}}$ showed comparable precision and recall.

[2] The ontology can be downloaded at http://www.aifb.uni-karlsruhe.de/WBS/pci/FinanceGoldStandard.isa. P. Cimiano and his colleagues added English labels for the originally German labeled nodes (Cimiano et al., 2005)

[3] http://www.daviddlewis.com/resources/testcollections/reuters21578/

[4] http://www.connexor.com/

We harmonized $Spec_{\mathrm{in}}$ and $Spec_{\mathrm{adj}}$ to $Spec_{\mathrm{in/adj}}$ as described in (Ryu & Choi, 2005) to take advantages of both inside and outside information. Harmonic mean of two specificity values was used in $Spec_{\mathrm{in/adj}}$ method. $Spec_{\mathrm{in/adj}}$ showed the highest F-measure because precision was higher than that of $Spec_{\mathrm{in}}$ and recall was equal to that of $Spec_{\mathrm{in}}$.

Table 1. Precision, recall and F-measure for term specificity

Method	Precision	Recall	F-measure
$Spec_{\mathrm{adj}}$	**0.795**	0.609	0.689
$Spec_{\mathrm{varg}}$	0.663	0.702	0.682
$Spec_{\mathrm{coldoc}}$	0.717	0.702	0.709
$Spec_{\mathrm{in}}$	0.728	**0.907**	0.808
$Spec_{\mathrm{in/adj}}$	**0.731**	**0.907**	**0.810**

5.2 Term Similarity

We evaluated similarity measures by comparing with taxonomy based similarity measure. (Budanitsky & Hirst, 2006) calculated correlation coefficients (CC) between human similarity ratings and the five WordNet based similarity measures. Among the five computational measures, (Leacock & Chodorow, 1998)'s method showed the highest correlation coefficients, even though all of the measures showed similar ranging from 0.74 to 0.85. This result means that taxonomy based similarity is highly correlated to human similarity ratings. We can indirectly evaluate our similarity measures by comparing to taxonomy based similarity measure, instead of direct comparison to human rating. If applied similarity measure is qualified, the calculated similarity will be highly correlated to taxonomy based similarity. Leacock and Chodorow proposed following formula for computing the scaled semantic similarity between terms t_1 and t_2 in taxonomy.

$$Sim_{LC}(t_1,t_2) = -\log \frac{len(t_1,t_2)}{2 \times \max_{t \in Taxonomy} depth(t)} \qquad (12)$$

where the denominator includes the maximum depth of given taxonomy, and $len(t_1, t_2)$ is number of edges in the shortest path between word t_1 and t_2 in the taxonomy.

Besides CC with ontology based similarity measures, recall of a similarity measures is also important evaluation factor. We defined recall of similarity measure, R_{Sim}, as the fraction of the term pairs that have similarity values by the given measuring method as Eq. (13).

$$R_{Sim} = \frac{\#\ similarity\ measured\ term\ pairs}{\#\ all\ possible\ term\ pairs} \qquad (13)$$

We also defined F-measure for a similarity measure, F_{sim}, as harmonic means of CC and R_{sim}. Because CC is a kind of precision, F_{sim} is overall measure of precision and recall.

We calculated term similarity between all possible term pairs in finance ontology using the measures described in section 3. Additionally we introduced new similarity measure $Sim_{\mathrm{in/varg}}$ which is combined similarity of Sim_{varg} and Sim_{in}. Sim_{varg} and Sim_{in} between two terms are harmonized to $Sim_{\mathrm{in/varg}}$. We also calculated Sim_{LC} based on finance ontology, and calculated CC between Sim_{LC} and results of other measures. Figure 5 shows variation of CC and recall as threshold of similarity changes from 0.0 to 1.0 for five similarity measures. Threshold is directly proportional to CC and inversely proportional to recall in ideal case. We normalized all similarity values to [0.0, 1.0] in each measure. CC grows as threshold increases in Sim_{coldoc} and Sim_{varg} as we expected. CC of CSM measure, Sim_{csm}, increased as threshold increased and decreased when threshold is over 0.6. For example two terms '*asset*' and '*current asset*' are very similar to each other based on Sim_{LC} measure, because edge count between two terms is one in finance ontology. The former can be modified many adjectives such as '*intangible*', '*tangible*', '*new*' and '*estimated*', while the latter is rarely modified by other adjectives in corpus because it was already extended from '*asset*' by adding adjective '*current*'. Therefore, semantically similar terms do not always have similar adjective distributions. CC between Sim_{in} and Sim_{LC} showed high curve in low threshold, but downed as threshold increased. Similarity value above 0.6 is insignificant, because it is hard to be over 0.6 using Eq. (10). For example, similarity between '*executive board meeting*' and '*board meeting*' is 0.8, the maximum similarity in our test set. The average of inside-word similarity is 0.41.

Sim_{varg} showed higher recall than other measures. This means that verb-argument relation is more abundant than other features in corpus. Sim_{In} showed the lowest recall because we could get valid similarity using Eq. (10). Sim_{varg} showed higher F-measure when threshold is over 0.2. This result illustrate that verb-argument relation is adequate feature to similarity calculation.

The combined similarity measure, $Sim_{\text{in/varg}}$, complement shortcomings of Sim_{In} and Sim_{varg}. Sim_{In} showed high CC but low recall. Contrarily Sim_{varg} showed low CC but high recall. $Sim_{\text{in/varg}}$ showed the highest F-measure.

5.3 Taxonomy learning

In order to evaluate our approach we need to assess how good the automatically learned taxonomies reflect a given domain. The goodness is evaluated by the similarity of automatically learned taxonomy to reference taxonomy. We used (Cimiano et al., 2005)'s ontology evaluation method in which lexical recall (LR_{Tax}), precision (P_{Tax}) and F-measure (F_{Tax}) of learned taxonomy are defined based on the notion of taxonomy overlap. LR_{Tax} is defined as the ratio of number of common terms in learned taxonomy and reference taxonomy over number of terms in reference taxonomy. P_{Tax} is defined as ratio of taxonomy overlap of learned taxonomy to reference taxonomy. F_{Tax} is harmonic mean of LR_{Tax} and P_{Tax}.

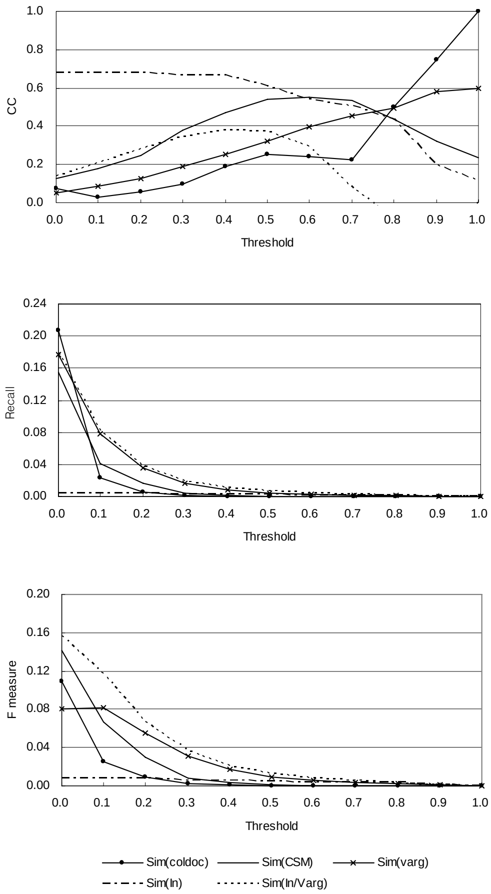


Figure 5 Correlation coefficient between Sim_{LC} and other similarity measures. Recall and F-measure of similarity measures

We generated four taxonomies, T_{coldoc}, T_{csm}, T_{fca}, $T_{\text{spec/sim}}$, using four taxonomy learning methods: term co-occurring method, CSM method, FCA method and our method. We applied $Spec_{\text{in/adj}}$ in specificity measuring and $Sim_{\text{in/varg}}$ in similarity calculation because they showed the highest F-measure. In our method, the most probable one term was selected as hypernym of newly inserted term in each learning step.

Figure 6 shows variations of lexical recall, precision and F-measure of four methods as threshold changes. Threshold in each method represent different information to each other. Threshold in T_{csm} is variation of CSM values. Threshold in T_{coldoc} is variation of probability of two terms co-occur in a document. Threshold in T_{fca} is normalized frequency of contexts. Threshold in $T_{\text{spec/sim}}$, is variation of similarity.

$T_{\text{spec/sim}}$ showed the highest lexical recall. Lexical recall is tightly related to recall in similarity measures. $Sim_{\text{in/varg}}$ showed the highest recall in similarity measures. T_{fca} and T_{csm} showed higher precision than other taxonomies. It is assumed that precision of taxonomy depends on

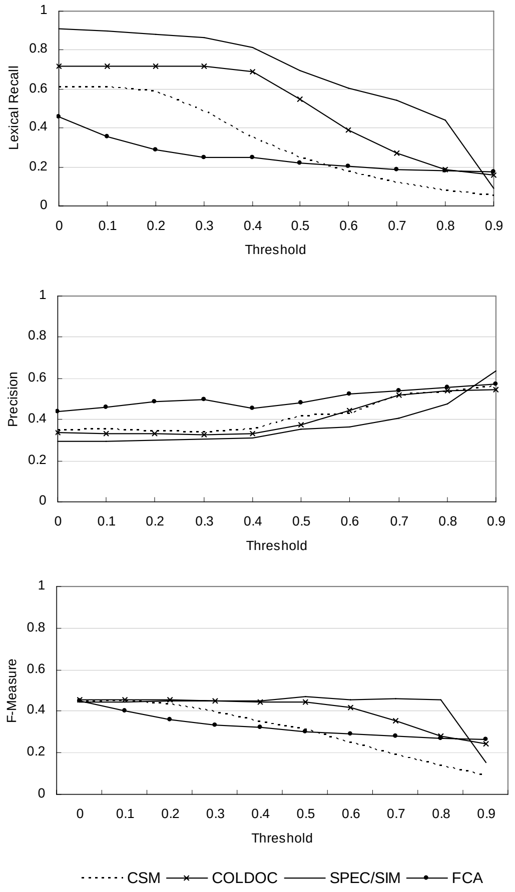


Figure 6. Lexical recall, precision and F-measure of taxonomy learning methods

the precision of specificity measures and the CC of similarity measures. In actual case, Sim_{varg} showed the most plausible curve in CC and $Spec_{\mathrm{adj}}$ showed the highest precision in specificity. Verb-argument relation and adjective-term relation are used in FCA and CSM methods respectively. $T_{\mathrm{spec/sim}}$ and T_{coldoc} showed higher F-measure curve than other two taxonomies due to high lexical recall. Although our method showed plausible F-measure, it showed the lowest precision. So other combination of similarity and specificity measures are needed to improve precision of learned taxonomy.

6 Conclusion

We have presented new taxonomy learning method with term similarity and specificity taken from domain-specific corpus. It can be applied to different domains as it is; and, if we have a syntactic parser available, to different languages. We analyzed the features used in previous researches in view of term specificity and similarity. In this analysis, we found that the features embed the characteristics of both conditions.

Compared to previous approaches, our method has advantages in that we can use different features for term specificity and similarity. It makes easy to analyze errors in taxonomy learning step, whether the wrong relations are caused by specificity errors or by similarity errors. The main drawback of our method, as it is now, is that the effect of wrong located terms in upper level propagates to lower levels.

Until now, researches on automatic ontology learning especially taxonomic relation showed very low precision. Human experts' intervention is inevitable in automatic learning process to make applicable taxonomy. Future work is to make new model where human experts and system work interactively in ontology learning process in order to balance cost and precision.

Reference

S. Caraballo, E. Charniak. 1999. Determining the Specificity of Nouns from Text. *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 63-70

P. Cimiano, A. Hotho, S.Staab. 2005. Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. *Journal of AI Research*, Vol. 24, pp. 305-339

M. Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. *Proceedings of the 14th International Conference on Computational Linguistics*

L. Iwanska, N. Mata and K. Kruger. 2000. Fully automatic acquisition of taxonomic knowledge from large corpora of texts. In Iwanska, L. & Shapiro, S. (Eds.), *Natural Language Processing and Knowledge Processing*, pp. 335-345, MIT/AAAI Press.

E. Yamamoto, K. Kanzaki and H. Isahara. 2005. Extraction of Hierarchies Based on Inclusion of Co-occurring Words with Frequency Information. *Proceedings of 9th International Joint Conference on Artificial Intelligence*, pp. 1160-1167

A. Burgun, O. Bodenreider. 2001. Aspects of the Taxonomic Relation in the Biomedical Domain, *Proceedings of International Conference on Formal Ontology in Information Systems*, pp. 222-233

Mark Sanderson and Bruce Croft. 1999. Deriving concept hierarchies from text. *Proceedings of the 22th Annual ACM S1GIR Conference on Research and Development in Information Retrieval*, pp. 206-213, 1999

Karen Sparck Jones. 1972. Exhausitivity and Specificity *Journal of Documentation* Vol. 28, Num. 1, pp. 11-21

S.K.M. Wong, Y.Y. Yao. 1992. An Information-Theoretic Measure of Term Specificity, *Journal of the American Society for Information Science*, Vol. 43, Num. 1. pp.54-61

ISO 704. 2000. *Terminology work-Principle and methods*. ISO 704 Second Edition

A. Aizawa. 2003. An information-theoretic perspective of tf-idf measures. *Journal of Information Processing and Management*, vol. 39

Alexander Budanitsky, Graeme Hirst. 2006 Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*. Vol. 32 NO. 1, pp. 13-47(35)

Claudia Leacock, Martin Chodorow. 1998. Combining local context and WordNet similarity for word sense identification. In Christian Fellbaum, editor, *WordNet: An Electronic Lexical Database*. The MIT Press, pp. 265-283

Pum-Mo Ryu, Key-Sun Choi. 2005. An Information-Theoretic Approach to Taxonomy Extraction for Ontology Learning, In P. Buitelaar et al. (eds.), *Ontology Learning from Text: Methods, Evaluation and Applications*, Vol. 123, Frontiers in Artificial Intelligence and Applications, IOS Press

Farid Cerbah. 2000. Exogeneous and Endogeneous Approaches to Semantic Categorization of Unknown Technical Terms. *Proceedings of the 18th International Conference on Computational Linguistics*, vol. 1, pp. 145-151

Towards Large-scale Non-taxonomic Relation Extraction: Estimating the Precision of Rote Extractors*

Enrique Alfonseca*† **Maria Ruiz-Casado***† **Manabu Okumura*** **Pablo Castells**†

*Precision and Intelligence Laboratory
Tokyo Institute of Techonology
enrique@lr.pi.titech.ac.jp
oku@pi.titech.ac.jp

†Computer Science Department
Universidad Autonoma de Madrid
enrique.alfonseca@uam.es
maria.ruiz@uam.es
pablo.castells@uam.es

Abstract

In this paper, we describe a rote extractor that learns patterns for finding semantic relations in unrestricted text, with new procedures for pattern generalisation and scoring. An improved method for estimating the precision of the extracted patterns is presented. We show that our method approximates the precision values as evaluated by hand much better than the procedure traditionally used in rote extractors.

1 Introduction

With the large growth of the information stored in the web, it is necessary to have available automatic or semi-automatic tools so as to be able to process all this web content. Therefore, a large effort has been invested in developing automatic or semi-automatic techniques for locating and annotating patterns and implicit information from the web, a task known as Web Mining. In the particular case of web content mining, the aim is automatically mining data from textual web documents that can be represented with machine-readable semantic formalisms such as ontologies and semantic-web languages.

Recently, there is an increasing interest in automatically extracting structured information from large corpora and, in particular, from the Web (Craven et al., 1999). Because of the characteristics of the web, it is necessary to develop efficient algorithms able to learn from unannotated data (Riloff and Schmelzenbach, 1998; Soderland, 1999; Mann and Yarowsky, 2005). New types of web content such as blogs and wikis, are also a source of textual information that contain an underlying structure from which specialist systems can benefit.

Consequently, rote extractors (Brin, 1998; Agichtein and Gravano, 2000; Ravichandran and Hovy, 2002) have been identified as an appropriate method to look for textual contexts that happen to convey a certain relation between two concepts. In this paper, we describe a new procedure for estimating the precision of the patterns learnt by a rote extractor, and how it compares to previous approaches. The solution proposed opens new possibilities for improving the precision of the generated patterns, as described below.

This paper is structured as follows: Section 2 describe related work; Section 3 and 4 describe the proposed procedure and its evaluation, and Section 5 presents the conclusions and future work.

2 Related work

Extracting information using Machine Learning algorithms has received much attention since the nineties, mainly motivated by the Message Understanding Conferences. From the mid-nineties, there are systems that learn extraction patterns from partially annotated and unannotated data (Huffman, 1995; Riloff, 1996; Riloff and Schmelzenbach, 1998; Soderland, 1999).

Generalising textual patterns (both manually and automatically) for the identification of relations has been proposed since the early nineties (Hearst, 1992), and it has been applied to extending ontologies with hyperonymy and holonymy relations (Morin and Jacquemin, 1999; Kietz et al., 2000; Cimiano et al., 2004; Berland and Charniak, 1999). Finkelstein-Landau and Morin (1999) learn patterns for company merging relations with exceedingly good accuracies. Recently, kernel

*This work has been sponsored by MEC, project number TIN-2005-06885.

Proceedings of the 2nd Workshop on Ontology Learning and Population, pages 49–56,
Sydney, July 2006. ©2006 Association for Computational Linguistics

methods are also becoming widely used for relation extraction (Bunescu and Mooney, 2005; Zhao and Grishman, 2005).

Concerning rote extractors from the web, they have the advantage that the training corpora can be collected easily and automatically, so they are useful in discovering many different relations from text. Several similar approaches have been proposed (Brin, 1998; Agichtein and Gravano, 2000; Ravichandran and Hovy, 2002), with various applications: Question-Answering (Ravichandran and Hovy, 2002), multi-document Named Entity Coreference (Mann and Yarowsky, 2003), and generating biographical information (Mann and Yarowsky, 2005). Szpektor et al. (2004) applies a similar, with no seed lists, to extract automatically entailment relationships between verbs, and Etzioni et al. (2005) report very good results extracting Named Entities and relationships from the web.

2.1 Rote extractors

Rote extractors (Mann and Yarowsky, 2005) estimate the probability of a relation $r(p, q)$ given the surrounding context $A_1pA_2qA_3$. This is calculated, with a training corpus T, as the number of times that two related elements $r(x, y)$ from T appear with that same context $A_1xA_2yA_3$, divided by the total number of times that x appears in that context together with any other word:

$$P(r(p,q)|A_1pA_2qA_3) = \frac{\sum_{x,y\epsilon r} c(A_1xA_2yA_3)}{\sum_{x,z} c(A_1xA_2zA_3)} \qquad (1)$$

x is called the *hook*, and y the *target*. In order to train a Rote extractor from the web, this procedure is mostly used (Ravichandran and Hovy, 2002):

1. Select a pair of related elements to be used as seed. For instance, *(Dickens,1812)* for the relation *birth year*.
2. Submit the query *Dickens AND 1812* to a search engine, and download a number of documents to build the training corpus.
3. Keep all the sentences containing both elements.
4. Extract the set of contexts between them and identify repeated patterns. This may just be the m characters to the left or to the right (Brin, 1998), the longest common substring of several contexts (Agichtein and Gravano, 2000), or all substrings obtained with a suffix tree constructor (Ravichandran and Hovy, 2002).
5. Download a separate corpus, called *hook corpus*, containing just the hook (in the example, *Dickens*).
6. Apply the previous patterns to the hook corpus, calculate the precision of each pattern in the following way: the number of times it identifies a target related to the hook divided by the total number of times the pattern appears.
7. Repeat the procedure for other examples of the same relation.

To illustrate this process, let us suppose that we want to learn patterns to identify birth years. We may start with the pair (*Dickens*, *1812*). From the downloaded corpus, we extract sentences such as

Dickens was born in 1812
Dickens (1812 - 1870) was an English writer
Dickens (1812 - 1870) wrote Oliver Twist

The system identifies that the contexts of the last two sentences are very similar and chooses their longest common substring to produce the following patterns:

```
<hook> was born in <target>
<hook> ( <target> - 1870 )
```

The rote extractor needs to estimate automatically the precision of the extracted patterns, in order to keep the best ones. So as to measure these precision values, a hook corpus is now downloaded using the hook *Dickens* as the only query word, and the system looks for appearances of the patterns in this corpus. For every occurrence in which the hook of the relation is Dickens, if the target is 1812 it will be deemed correct, and otherwise it will be deemed incorrect (e.g. in *Dickens was born in Portsmouth*).

3 Our proposal

3.1 Motivation

In a rote extractor as described above, we believe that the procedure for calculating the precision of the patterns may be unreliable in some cases. For example, the following patterns are reported by Ravichandran and Hovy (2002) for identifying the relations Inventor, Discoverer and Location:

Relation	Prec.	Pattern
Inventor	1.0	<target> 's <hook> and
Inventor	1.0	that <target> 's <hook>
Discoverer	0.91	of <target> 's <hook>
Location	1.0	<target> 's <hook>

In the particular application in which they are used (relation extraction for Question Answering), they are useful because there is initially a question to be answered that indicates whether we are looking for an invention, a discovery or a location. However, if we want to apply them to unrestricted relation extraction, we have the problem that the same pattern, the genitive construction, represents all these relations, apart from the most common use indicating possession.

If patterns like these are so ambiguous, then why do they receive so high a precision estimate? One reason is that the patterns are only evaluated for the same hook for which they were extracted. To illustrate this with an example, let us suppose that we obtain a pattern for the relation *located-at* using the pairs *(New York, Chrysler Building)*. The genitive construction can be extracted from the context *New York's Chrysler Building*. Afterwards, when estimating the precision of this pattern, only sentences containing *<target>'s Chrysler Building* are taken into account. Because of this, most of the pairs extracted by this pattern may extract the target *New York*, apart from a few that extract the name of the architect that built it, *van Allen*. Thus we can expect that the genitive pattern will receive a high precision estimate as a *located-at* pattern.

For our purposes, however, we want to collect patterns for several relations such as *writer-book*, *painter-picture*, *director-film*, *actor-film*, and we want to make sure that the obtained patterns are only applicable to the desired relation. Patterns like *<target> 's <hook>* are very likely to be applicable to all of these relations at the same time, so we would like to be able to discard them automatically by assigning them a low precision.

3.2 Suggested improvements

Therefore, we propose the following three improvements to this procedure:

1. Collecting not only a *hook corpus* but also a *target corpus* should help in calculating the precision. In the example of the *Chrysler building*, we have seen that in most cases that we look for the pattern *'s Chrysler building* the previous words are *New York*, and so the pattern is considered accurate. However, if we look for the pattern *New York's*, we shall surely find it followed by many different terms representing different relations, and the precision estimate will decrease.
2. Testing the patterns obtained for one relation using the hook and target corpora collected for other relations. For instance, if the genitive construction has been extracted as a possible pattern for the *writer-book* relation, and we apply it to a corpus about painters, the rote extractor can detect that it also extracts pairs with painters and paintings, so that particular pattern will not be very precise for that relation.
3. Many of the pairs extracted by the patterns in the hook corpora were not evaluated at all when the hook in the extracted pair was not present in the seed lists. To overcome this, we propose to use the web to check whether the extracted pair might be correct, as shown below.

3.3 Algorithm

In our implementation, the rote extractor starts with a table containing some information about the relations for which we want to learn patterns. This procedure needs a little more information than just the seed list, which is provided as a table in the format displayed in Table 1. The data provided for each relation is the following: (a) The **name of the relation**, used for naming the output files containing the patterns; (b) the name of the file containing the **seed list**; (c) the cardinality of the relation. For instance, given that many people can be born on the same year, but for every person there is just one birth year, the cardinality of the relation *birth year* is n:1; (d) the **restrictions** on the hook and the target. These can be of the following three categories: *unrestricted*, if the pattern can extract any sequence of words as hook or target of the relation, *Entity*, if the pattern can extract as hook or target only things of the same entity type as the words in the seed list (as annotated by the NERC module), or *PoS*, if the pattern can extract as hook or target any sequence of words whose sequence of PoS labels was seen in the training corpus; and (e) a sequence of **queries** that could be used to check, using the web, whether an extracted pair is correct or not.

We assume that the system has used the seed list to extract and generalise a set of patterns for each of the relations using training corpora (Ravichandran and Hovy, 2002; Alfonseca et al., 2006a). Our procedure for calculating the patterns' precisions is as follows:

1. For every relation,
    (a) For every *hook*, collect a *hook corpus* from the web.

Relation name	Seed-list	Cardinality	Hook-type	Target-type	Web queries
birth year	birth-date.txt	n:1	entity	entity	\$1 was born in \$2
death year	death-date.txt	n:1	entity	entity	\$1 died in \$2
birth place	birth-place.txt	n:1	entity	entity	\$1 was born in \$2
country-capital	country-capital.txt	1:1	entity	entity	\$2 is the capital of \$1
author-book	author-book.txt	n:n	entity	unrestricted	\$1 is the author of \$2
director-film	director-film.txt	1:n	entity	unrestricted	\$1 directed \$2, \$2 directed by \$1

Table 1: Example rows in the input table for the system.

(b) For every *target*, collect a *target corpus* from the web.

2. For every relation r,

   (a) For every pattern P, collected during training, apply it to every hook and target corpora to extract a set of pairs.

   For every pair $p = (p_h, p_t)$,

   - If it appears in the seed list of r, consider it correct.
   - If it appears in the seed list of other relation, consider it incorrect.
   - If the hook p_h appears in the seed list of r with a different target, and the cardinality is 1:1 or n:1, consider it incorrect.
   - If the target p_t appears in r's seed list with a different hook, and the cardinality is 1:1 or 1:n, incorrect.
   - Otherwise, the seed list does not provide enough information to evaluate p, so we perform a test on the web. For every query provided for r, the system replaces \$1 with p_h and \$2 with p_t, and sends the query to Google. The pair is deemed correct if and only if there is at least one answer.

   The precision of P is estimated as the number of extracted pairs that are supposedly correct divided by the total number of pairs extracted.

In this step, every pattern that did not apply at least twice in the hook and target corpora is also discarded.

3.4 Example

After collecting and generalising patterns for the relation *director-film*, we apply each pattern to the hook and target corpora collected for every relation. Let us suppose that we want to estimate the precision of the pattern

<target> 's <hook>

and we apply it to the hook and the target corpora for this relation and for *author-book*. Possible pairs extracted are *(Woody Allen, Bananas), (Woody Allen, Without Fears), (Charles Dickens, A Christmas Carol)*. Only the first one is correct. The rote extractor proceeds as follows:

- The first pair appears in the seed list, so it is considered correct.
- Although *Woody Allen* appears as hook in the seed list and *Without Fears* does not appear as target, the second pair is still not considered incorrect because the *directed-by* relation has n:n cardinality.
- The third pair appears in the seed list for *writer-book*, so it is directly marked as incorrect.
- Finally, because still the system has not made a decision about the second pair, it queries Google with the sequences

  Woody Allen directed Without Fears

  Without Fears directed by Woody Allen

  Because neither of those queries provide any answer, it is considered incorrect.

In this way, it can be expected that the patterns that are equally applicable to several relations, such as *writer-book*, *director-film* or *painter-picture* will attain a low precision because they will extract many incorrect relations from the corpora corresponding to the other relations.

4 Experiment and results

4.1 Rote extractor settings

The initial steps of the rote extractor follows the general approach: downloading a training corpus using the seed list and extracting patterns. The training corpora are processed with a part-of-speech tagger and a module for Named Entity Recognition and Classification (NERC) that annotates people, organisations, locations, dates, relative temporal expressions and numbers (Alfonseca et al., 2006b), so this information can be included in the patterns. Furthermore, for each of the terms in a pair in the training corpora, the system also

Birth year:

```
BOS/BOS <hook> (/( <target> -/- number/entity )/) EOS/EOS
BOS/BOS <hook> (/( <target> -/- number/entity )/) British/JJ writer/NN
BOS/BOS <hook> was/VBD born/VBN on/IN the/DT first/JJ of/IN time_expr/entity ,/, <target> ,/, at/IN location/entity ,/, of/IN
BOS/BOS <hook> (/( <target> -/- )/) a/DT web/NN guide/NN
```

Birth place:

```
BOS/BOS <hook> was/VBD born/VBN in/IN <target> ,/, in/IN central/JJ location/entity ,/,
BOS/BOS <hook> was/VBD born/VBN in/IN <target> date/entity and/CC moved/VBD to/TO location/entity
BOS/BOS Artist/NN :/, <hook> -/- <target> ,/, location/entity (/( number/entity -/-
BOS/BOS <hook> ,/, born/VBN in/IN <target> on/IN date/entity ,/, worked/VBN as/IN
```

Author-book:

```
BOS/BOS <hook> author/NN of/IN <target> EOS/EOS
BOS/BOS Odysseus/NNP :/, Based/VBN on/IN <target> ,/, <hook> ’s/POS epic/NN from/IN Greek/JJ mythology/NN
BOS/BOS Background/NN on/IN <target> by/IN <hook> EOS/EOS
did/VBD the/DT circumstances/NNS in/IN which/WDT <hook> wrote/VBD "/’’ <target> "/’’ in/IN number/entity ,/, and/CC
```

Capital-country:

```
BOS/BOS <hook> is/VBZ the/DT capital/NN of/IN <target> location/entity ,/, location/entity correct/JJ time/NN
BOS/BOS The/DT harbor/NN in/IN <hook> ,/, the/DT capital/NN of/IN <target> ,/, is/VBZ number/entity of/IN location/entity
BOS/BOS <hook> ,/, <target> EOS/EOS
BOS/BOS <hook> ,/, <target> -/- organization/entity EOS/EOS
```

Figure 1: Example patterns extracted from the training corpus for each several kinds of relations.

stores in a separate file the way in which they are annotated in the training corpus: the sequences of part-of-speech tags of every appearance, and the entity type (if marked as such). So, for instance, typical PoS sequences for names of authors are “NNP”[1] (surname) and “NNP NNP” (first name and surname). A typical entity kind for an author is person.

In the case that a pair from the seed list is found in a sentence, a context around the two words in the pair is extracted, including (a) at most five words to the left of the first word; (b) all the words in between the pair words; (c) at most five words to the right of the second word. The context never jumps over sentence boundaries, which are marked with the symbols BOS (*Beginning of sentence*) and EOS (*End of sentence*). The two related concepts are marked as <hook> and <target>. Figure 1 shows several example contexts extracted for the relations *birth year*, *birth place*, *writer-book* and *country-capital city*.

The approach followed for the generalisation is the one described by (Alfonseca et al., 2006a; Ruiz-Casado et al., in press), which has a few modifications with respect to Ravichandran and Hovy (2002)’s, such as the use of the wildcard * to represent any sequence of words, and the addition of part-of-speech and Named Entity labels to the patterns.

The input table has been built with the following nineteen relations: birth year, death year, birth place, death place, author–book, actor–film, director–film, painter–painting, Employee–organisation, chief of state, soccer player–team, soccer team-city, soccer team-manager, country or region–capital city, country or region–area, country or region–population, country–bordering country, country-name of inhabitant (e.g. Spain-Spaniard), and country-continent. The time required to build the table and the seed lists was less than one person-day, as some of the seed lists were directly collected from web pages.

Relation	Seeds	Extr.	Gener.	Filt.
Birth year	244	2374	4748	30
Death year	216	2178	4356	14
Birth place	169	764	1528	28
Death place	76	295	590	6
Author-book	198	8297	16594	283
Actor-film	49	739	1478	3
Director-film	85	6933	13866	200
Painter-painting	92	597	1194	15
Employee-organisation	62	1667	3334	6
Chief of state	55	1989	3978	8
Soccer player-team	194	4259	8518	39
Soccer team-city	185	180	360	0
Soccer team-manager	43	994	1988	9
Country/region-capital city	222	4533	9066	107
Country/region-area	226	762	1524	2
Country/region-population	288	318	636	3
Country-bordering country	157	6828	13656	240
Country-inhabitant	228	2711	5422	17
Country-continent	197	1606	3212	21

Table 2: Number of seed pairs for each relation, and number of unique patterns in each step.

For each step, the following settings have been set:

- The size of the training corpus has been set to 50 documents for each pair in the original seed lists. Given that the typical sizes of the lists collected are between 50 and 300 pairs, this means that several thousand documents are downloaded for each relation.
- Before the generalisation step, the rote extractor discards those patterns in which the hook and the target are too far away to each other, because they are usually difficult to generalise. The maximum allowed distance

[1] All the PoS examples in this paper are done with Penn Treebank labels.

No.	Pattern	Applied	Prec1	Prec2	Real
1	Biography|Hymns|Infography|Life|Love|POETRY|Poetry|Quotations|Search|Sketch|Woolf|charts|genius|kindness|poets/NN */* OF|Of|about|by|for|from|like|of/IN <hook> (/(<target> -/-	6	1.00	1.00	1.00
2	"/'' <hook> (/(<target> -/-	4	1.00	1.00	1.00
3	[BOS]/[BOS] <hook> was/VBD born/VBN about|around|in/IN <target> B.C.|B.C.E|BC/NNP at|in/IN	3	1.00	1.00	1.00
4	[BOS]/[BOS] <hook> was/VBD born/VBN about|around|in/IN <target> B.C.|B.C.E|BC/NNP at|in/IN location/entity	3	1.00	1.00	1.00
5	[BOS]/[BOS] <hook> was/VBD born/VBN around/IN <target> B.C.E/NNP at/IN location/entity ,/, a/DT	3	1.00	1.00	1.00
6	[BOS]/[BOS] <hook> was/VBD born/VBN around|in/IN <target> B.C.|B.C.E/NNP at|in/IN location/entity ,/,	3	1.00	1.00	1.00
7	[BOS]/[BOS] */* ATTRIBUTION|Artist|Author|Authors|Composer|Details|Email|Extractions|Myth|PAL|Person|Quotes|Title|Topic/NNP :/, <hook> (/(<target> -/-	3	1.00	1.00	1.00
8	classical/JJ playwrights/NNS of/IN organisation/entity ,/, <hook> was/VBD born/VBN near/IN location/entity in/IN <target> BCE/NNP ,/, in/IN the/DT village/NN	3	1.00	1.00	1.00
9	[BOS]/[BOS] <hook> (/(<target> -/-)/)	2	1.00	1.00	1.00
10	[BOS]/[BOS] <hook> (/(<target> -|--/-)/)	2	1.00	1.00	1.00
11	[BOS]/[BOS] <hook> (/(<target> person/entity BC/NNP ;/, Greek/NNP :/,	2	1.00	1.00	1.00
12	ACCESS|AND|Alice|Author|Authors|BY|Biography|CARL|Dame|Don|ELIZABETH|(...)|web|writer|writerMuriel|years/NNP <hook> (/(<target> -|- -/-	8	0.75	1.00	
13	-/- <hook> (/(<target> -/-	3	0.67	1.00	0.67
14	-|--/- <hook> (/(<target> -/-	3	0.67	1.00	0.67
15	[BOS]/[BOS] <hook> (/(<target> -/-	60	0.62	1.00	0.81
16	[BOS]/[BOS] <hook> (/(<target> -/- */*)/)	60	0.62	1.00	0.81
17	[BOS]/[BOS] <hook> (/(<target> -|--/-	60	0.62	1.00	0.81
18	,|:/, <hook> (/(<target> -/-	32	0.41	0.67	0.28
19	[BOS]/[BOS] <hook> ,/, */* (/(<target> -|--/-	15	0.40	1.00	0.67
20	,|:|;/, <hook> (/(<target> -|--/-	34	0.38	0.67	0.29
21	AND|Alice|Authors|Biography|Dame|Don|ELIZABETH|Email|Fiction|Frances|GEORGE|Home|I.|Introduction|Jean|L|Neben|PAL|PAULA|Percy|Playwrights|Poets|Sir|Stanisaw|Stanislaw|W.|WILLIAM|feedback|history|writer/NNP <hook> (/(<target> -/-	3	0.33	n/a	0.67
22	AND|Frances|Percy|Sir/NNP <hook> (/(<target> -/-	3	0.33	n/a	0.67
23	Alice|Authors|Biography|Dame|Don|ELIZABETH|Email|Fiction|Frances|GEORGE|Home|I.|Introduction|Jean|L|Neben|PAL|PAULA|Percy|Playwrights|Poets|Sir|Stanisaw|Stanislaw|W.|WILLIAM|feedback|history|writer/NN <hook> (/(<target> -/-	3	0.33	n/a	0.67
24	[BOS]/[BOS] <hook> ,|:/, */* ,|:/, <target> -/-	7	0.28	0.67	0.43
25	[BOS]/[BOS] <hook> ,|:/, <target> -/-	36	0.19	1.00	0.11
26	[BOS]/[BOS] <hook> ,/, */* (/(<target>)/)	20	0.15	0.33	0.10
27	[BOS]/[BOS] <target> <hook> ,/,	18	0.00	n/a	0.00
28	In|On|on/IN <target> ,/, <hook> grew|was/VBD	17	0.00	0.00	0.00
29	In|On|on/IN <target> ,/, <hook> grew|was|went/VBD	17	0.00	0.00	0.00
30	[BOS]/[BOS] <hook> ,/, */* DE|SARAH|VON|dramatist|novelist|playwright|poet/NNP (/(<target> -/-	3	0.00	n/a	1.0
	TOTAL	436	0.46	0.84	0.54

Table 3: Patterns for the relation *birth year*, results extracted by each, precision estimated with this procedure and with the traditional hook corpus approach, and precision evaluated by hand).

between them has been set to 8 words.

- At each step, the two most similar patterns are generalised, and their generalisation is added to the set of patterns. No pattern is discarded at this step. This process stops when all the patterns resulting from the generalisation of existing ones contain wildcards adjacent to either the hook or the target.
- For the precision estimation, for each pair in the seed lists, 50 documents are collected for the hook and other 50 for the target. Because of time constraints, and given that the total size of the hook and the target corpora exceeds 100,000 documents, for each pattern a sample of 250 documents is randomly chosen and the patterns are applied to it. This sample is built randomly but with the following constraints: there should be an equal amount of documents selected from the corpora from each relationship; and there should be an equal amount of documents from hook corpora and from target corpora.

4.2 Output obtained

Table 2 shows the number of patterns obtained for each relation. Note that the generalisation procedure applied produces new (generalised) patterns to the set of original patterns, but no original pattern is removed, so they all are evaluated; this is why the set of patterns increases after the generalisation. The filtering criterion was to keep the patterns that applied at least twice on the test corpus.

It is interesting to see that for most relations the reduction of the pruning is very drastic. This is because of two reasons: Firstly, most patterns are far too specific, as they include up to 5 words at each side of the hook and the target, and all the words in between. Only those patterns that have generalised very much, substituting large portions with wildcards or disjunctions are likely to apply to the sentences in the hook and target corpora.

Secondly, the samples of the hook and target corpora used are too small for some of the relations to apply, so few patterns apply more than twice.

Note that, for some relations, the output of the generalisation step contains less patterns that the output of the initial extraction step: that is due to the fact that the patterns in which the hook and the target are not nearby were removed in between these two steps.

Concerning the precision estimates, a full evaluation is provided for the *birth-year* relation. Table 3 shows in detail the thirty patterns obtained. It can also be seen that some of the patterns with good precision contain the wildcard *. For instance, the first pattern indicates that the presence of any of the words *biography*, *poetry*, etc. anywhere in a sentence before a person name and a date or number between parenthesis is a strong indication that the target is a birth year.

The last columns in the table indicate the number of times that each rule applied in the hook and target corpora, and the precision of the rule in each of the following cases:

- As estimated by the complete program (Prec1).
- As estimated by the traditional hook corpus approach (Prec2). Here, cardinality is not taken into account, patterns are evaluated only on the hook corpora from the same relation, and those pairs whose hook is not in the seed list are ignored.
- The real precision of the rule (real). In order to obtain this metric, two different annotators evaluated the pairs applied independently, and the precision was estimated from the pairs in which they agreed (there was a 96.29% agreement, Kappa=0.926).

As can be seen, in most of the cases our procedure produces lower precision estimates.

If we calculate the total precision of all the rules altogether, shown in the last row of the table, we can see that, without the modifications, the whole set of rules would be considered to have a total precision of 0.84, while that estimate decreases sharply to 0.46 when they are used. This value is nearer the precision of 0.54 evaluated by hand. Although it may seem surprising that the precision estimated by the new procedure is even lower than the real precision of the patterns, as measured by hand, that is due to the fact that the web queries consider unknown pairs as incorrect unless they appear in the web exactly in the format of the query in the input table. Specially for not very well-known people, we cannot expect that all of them will appear in the web following the pattern *"X was born in date"*, so the web estimates tend to be over-conservative.

Relation	Prec1	Prec2	Real
Birth year	**0.46 [0.41,0.51]**	0.84 [0.81,0.87]	0.54 [0.49,0.59]
Death year	**0.29 [0.24,0.34]**	**0.55 [0.41,0.69]**	0.38 [0.31,0.44]
Birth place	0.65 [0.62,0.69]	0.36 [0.29,0.43]	0.84 [0.79,0.89]
Death place	0.82 [0.73,0.91]	1.00 [1.00,1.00]	0.96 [0.93,0.99]
Author-book	0.07 [0.07,0.07]	0.26 [0.19,0.33]	0.03 [0.00,0.05]
Actor-film	**0.07 [0.01,0.13]**	1.00 [1.00,1.00]	0.02 [0.00,0.03]
Director-film	0.03 [0.03,0.03]	0.26 [0.18,0.34]	0.01 [0.00,0.01]
Painter-painting	0.10 [0.07,0.12]	0.35 [0.23,0.47]	0.17 [0.12,0.22]
Employee-organisation	**0.31 [0.22,0.40]**	1.00 [1.00,1.00]	0.33 [0.26,0.40]
Chief of state	**0.00 [0.00,0.00]**	-	0.00 [0.00,0.00]
Soccer player-team	**0.07 [0.06,0.08]**	1.00 [1.00,1.00]	0.08 [0.04,0.12]
Soccer team-city	-	-	-
Soccer team-manager	0.61 [0.53,0.69]	1.00 [1.00,1.00]	0.83 [0.77,0.88]
Country/region-capital city	**0.12 [0.11,0.13]**	0.23 [0.22,0.24]	0.12 [0.07,0.16]
Country/region-area	**0.09 [0.00,0.19]**	1.00 [1.00,1.00]	0.06 [0.02,0.09]
Country/region-population	**1.00 [1.00,1.00]**	**1.00 [1.00,1.00]**	1.00 [1.00,1.00]
Country-bordering country	**0.17 [0.17,0.17]**	1.00 [1.00,1.00]	0.15 [0.10,0.20]
Country-inhabitant	**0.01 [0.00,0.01]**	0.80 [0.67,0.93]	0.01 [0.00,0.01]
Country-continent	0.16 [0.14,0.18]	0.07 [0.04,0.10]	0.00 [0.00,0.01]

Table 4: Precision estimates for the whole set of extracted pairs by *all* rules and all relations.

Table 4 shows the precision estimates for every pair extracted with all the rules using both procedures, with 0.95 confidence intervals. The real precision has been estimating by sampling randomly 200 pairs and evaluating them by hand, as explained above for the *birth year* relation. As can be observed, out of the 19 relations, the precision estimate of the whole set of rules for 11 of them is not statistically dissimilar to the real precision, while that only holds for two relationships using the previous approach.

Please note as well that the precisions indicated in the table refer to all the pairs extracted by all the rules, some of which are very precise, but some of which are very imprecise. If the rules are to be applied in an annotation system, only those with a high precision estimate would be used, and expectedly much better overall results would be obtained.

5 Conclusions and future work

We have described here a new procedure for estimating the precision of the patterns learnt by a rote extractor that learns from the web. Compared to other similar approaches, it has the following improvements:

- For each pair *(hook,target)* in the seed list, a *target corpora* is also collected (apart from the *hook corpora*), and the evaluation is performed using corpora from several relations.

This has been observed to improve the estimate of the rule's precision, given that the evaluation pairs not only refer to the elements in the seed list.

- The cardinality of the relations is taken into consideration in the estimation process using the seed list. This is important, for instance, to be able to estimate the precision in n:n relations like *author-work*, given that we cannot assume that the only books written by someone are those in the seed list.
- For those pairs that cannot be evaluated using the seed list, a simple query to the Google search engine is employed.

The precisions estimated with this procedure are significantly lower than the precisions obtained with the usual hook corpus approach, specially for ambiguous patterns, and much near the precision estimate when evaluated by hand.

Concerning future work, we plan to estimate the precision of the patterns using the whole hook and target corpora, rather than using a random sample. A second objective we have in mind is not to throw away the ambiguous patterns with low precision (e.g. the possessive construction), but to train a model so that we can disambiguate which is the relation they are conveying in each context (Girju et al., 2003).

References

E. Agichtein and L. Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of ICDL*, pages 85–94.

E. Alfonseca, P. Castells, M. Okumura, and M. Ruiz-Casado. 2006a. A rote extractor with edit distance-based generalisation and multi-corpora precision calculation. In *Poster session of ACL-2006*.

E. Alfonseca, A. Moreno-Sandoval, J. M. Guirao, and M. Ruiz-Casado. 2006b. The wraetlic NLP suite. In *Proceedings of LREC-2006*.

M. Berland and E. Charniak. 1999. Finding parts in very large corpora. In *Proceedings of ACL-99*.

S. Brin. 1998. Extracting patterns and relations from the World Wide Web. In *Proceedings of the WebDB Workshop at EDBT'98*.

R. Bunescu and R. J. Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the HLT Conference and EMNLP*.

P. Cimiano, S. Handschuh, and S. Staab. 2004. Towards the self-annotating web. In *Proceedings of the 13th World Wide Web Conference*, pages 462–471.

M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. 1999. Learning to construct knowledge bases from the world wide web. *Artificial Intelligence*, 118(1–2):69–113.

O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. 2005. Unsupervised named entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134.

M. Finkelstein-Landau and E. Morin. 1999. Extracting semantic relationships between terms: supervised vs. unsupervised methods. In *Workshop on Ontologial Engineering on the Global Info. Infrastructure*.

R. Girju, A. Badulescu, and D. Moldovan. 2003. Learning semantic constraints for the automatic discovery of part-whole relations. In *HLT-NAACL-03*.

M. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING-92*.

S. Huffman. 1995. Learning information extraction patterns from examples. In *IJCAI-95 Workshop on New Approaches to Learning for NLP*.

J. Kietz, A. Maedche, and R. Volz. 2000. A method for semi-automatic ontology acquisition from a corporate intranet. In *Workshop "Ontologies and text"*.

G. S. Mann and D. Yarowsky. 2003. Unsupervised personal name disambiguation. In *CoNLL-2003*.

G. S. Mann and D. Yarowsky. 2005. Multi-field information extraction and cross-document fusion. In *Proceedings of ACL 2005*.

E. Morin and C. Jacquemin. 1999. Projecting corpus-based semantic links on a thesaurus. In *ACL-99*.

D. Ravichandran and E. Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of ACL-2002*, pages 41–47.

E. Riloff and M. Schmelzenbach. 1998. An empirical approach to conceptual case frame acquisition. In *Proceedings of WVLC*, pages 49–56.

E. Riloff. 1996. Automatically generating extraction patterns from untagged text. In *AAAI*.

M. Ruiz-Casado, E. Alfonseca, and P. Castells. in press. Automatising the learning of lexical patterns: an application to the enrichment of WordNet by extracting semantic relationships from the Wikipedia. *Data and Knowledge Engineering*, in press.

S. Soderland. 1999. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1–3):233–272.

I. Szpektor, H. Tanev, I. Dagan, and B. Coppola. 2004. Scaling web-based acquisition of entailment relations. In *Proceedings of EMNLP 2004*.

S. Zhao and R. Grishman. 2005. Extracting relations with integrated information using kernel methods. In *Proceedings of ACL-2005*.

A hybrid approach for extracting semantic relations from texts

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 extracting relations from texts that exploits linguistic and empirical strategies, by means of a pipeline method involving a parser, part-of-speech tagger, named entity recognition system, pattern-based classification and word sense disambiguation models, and resources such as ontology, knowledge base and lexical databases. The relations extracted can be used for various tasks, including semantic web annotation and ontology learning. We suggest that the use of knowledge intensive strategies to process the input text and corpus-based techniques to deal with unpredicted cases and ambiguity problems allows to accurately discover the relevant relations between pairs of entities in that text.

1 Introduction

Semantic relations extracted from texts are useful for several applications, including question answering, information retrieval, semantic web annotation, and construction and extension of lexical resources and ontologies. In this paper we present an approach for relation extraction developed to semantically annotate relational knowledge coming from raw text, within a framework aiming to automatically acquire high quality semantic metadata for the Semantic Web.

In that framework, applications such as semantic web portals (Lei et al., 2006) analyze data from texts, databases, domain ontologies, and knowledge bases in order to extract the semantic knowledge in an integrated way. Known entities occurring in the text, i.e., entities that are included in the knowledge base, are semantically annotated with their properties, also provided by the knowledge base and by databases. New entities, as given by a named entity recognition system according to the possible types of entities in the ontology, are annotated without any additional information. In this context, the goal of the relation extraction approach presented here is to extract relational knowledge about entities, i.e., to identify the semantic relations between pairs of entities in the input texts. Entities can be both known and new, since named entity recognition is also carried out. Relations include those already existent in the knowledge base, new relations predicted as possible by the domain ontology, or completely new (unpredicted) relations.

The 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)[1] newsletters. We believe that by integrating corpus and knowledge-based techniques and using rich linguistic processing strategies in a completely automated fashion, the approach can achieve effective results, in terms of both accuracy and coverage.

With relational knowledge, a richer representation of the input data can be produced. Moreover, by identifying new entities, the relation extraction approach can also be applied to ontology population. Finally, since it extracts new relations, it can also be used as a first step for ontology learning.

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

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

Proceedings of the 2nd Workshop on Ontology Learning and Population, pages 57–64,
Sydney, July 2006. ©2006 Association for Computational Linguistics

our approach, showing its architecture and describing each of its main components (Section 3). Finally, we present the 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 detection of relations 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 (Yangarber, 2000; Stevenson, 2004).

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 (Miller et al, 2000), or clustering of semantically similar syntactic dependencies, according to their selectional restrictions (Gamallo et al., 2002).

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. (Roth and Yih, 2002), e.g., use 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, (Zelenko et al., 2003) use support vector machines and voted perceptron algorithms with a specialized kernel model. Also using kernel methods and support vector machines, (Zhao and Grishman, 2005) combine clues from different levels of syntactic information and applies composite kernels to integrate the individual kernels.

Similarly to our proposal, the framework presented by (Iria and Ciravegna, 2005) 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.

Several relation extraction approaches have been proposed focusing on the task of ontology learning (Reinberger and Spyns, 2004; Schutz and Buitelaar, 2005; Ciaramita et al., 2005). More comprehensive reviews can be found in (Maedche, 2002) and (Gomez-Perez and Manzano-Macho, 2003). These approaches aim to learn non-taxonomic relations between concepts, instead of lexical items. However, in essence, they can employ similar techniques to extract the relations. Additional strategies can be applied to determine whether the relations can be lifted from lexical items to concepts, as well as to determine the most appropriate level of abstraction to describe a relation (e.g. Maedche, 2002).

In the next section we describe our relation extraction 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.

3 A hybrid approach for relation extraction

The proposed approach for relation extraction is illustrated in Figure 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 terms linked by those relations. The detection 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 for the Semantic Web. Other potential applications include:

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

3) Ontology learning: new relations between existent concepts are identified, and can be used as a first step to extend an existent ontology. A subsequent step to lift relations between instances to an adequate level of abstraction may be necessary.

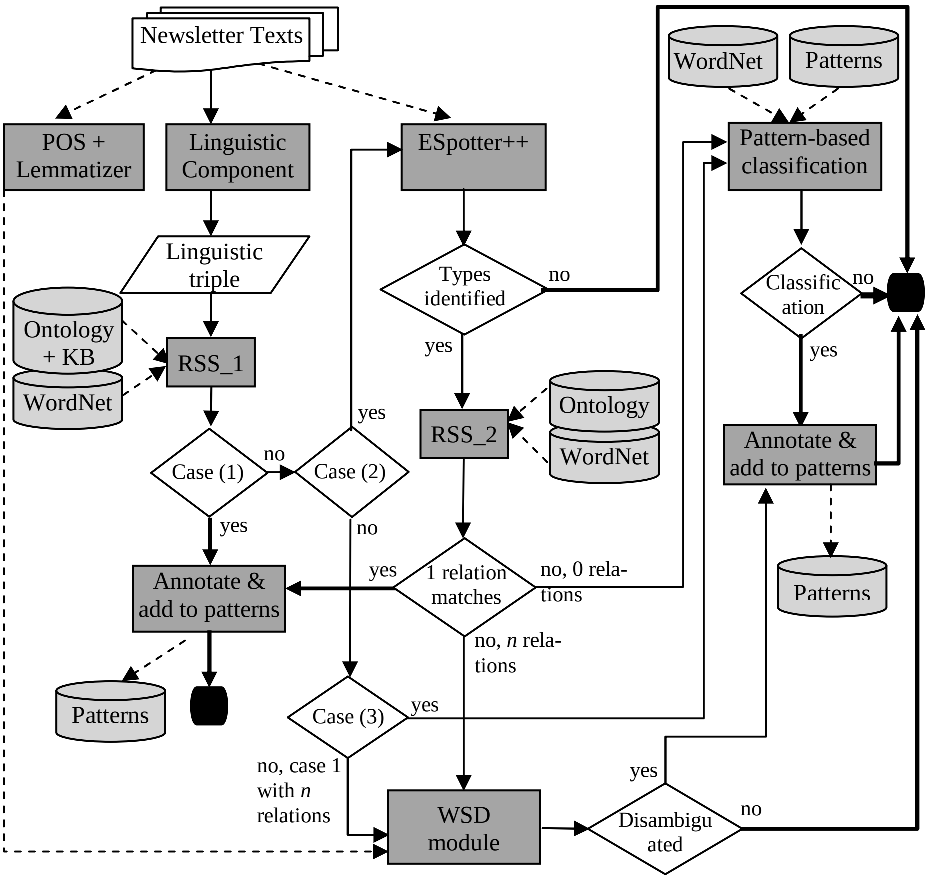


Figure 1. Architecture of the proposed approach

3.1 Context and resources

The input to our experiments consists of electronic **Newsletter Texts**[2]. These are short texts describing news of several natures related to members of a research group: projects, publications, events, awards, etc. The domain **Ontology** used (*KMi-basic-portal-ontology*) was designed based on the AKT reference ontology[3] to include concepts relevant to our 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** (Fellbaum, 1998) and a repository of **Patterns** of relations, described in Section 3.4.

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, and adaptation of the linguistic component designed in Aqualog (Lopez et al., 2005), a question answering system.

The linguistic component uses the infrastructure and the following resources from GATE (Cunningham et al., 2002): 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 (Cunningham et al., 2000), which allows the definition of patterns to recognize regular expressions using the annotations provided by GATE.

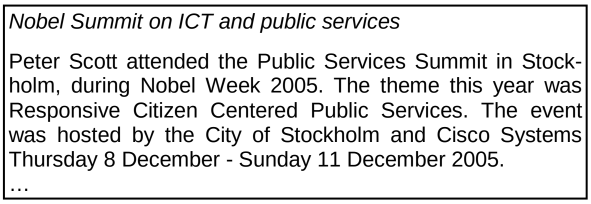
The main type of construction aimed to be identified by our grammar involves a verbal expression as indicative of a potential relation and two noun phrases as terms linked by that relation. However, our patterns also account for other types of constructions, including, e.g., the use of comma to implicitly indicate a relation, as in sentence (1). In this case, when mapping the terms into entities (Section 3.3), having identified that "KMi" is an *organization* and "Enrico

[2] http://news.kmi.open.ac.uk/kmiplanet/

[3] http://kmi.open.ac.uk/projects/akt/ref-onto/

Motta” is a *person*, it is possible to guess the relation indicated by the comma (e.g., work). Some examples triples identified by our patterns for the newsletter in Figure 2 are given in Figure 3.

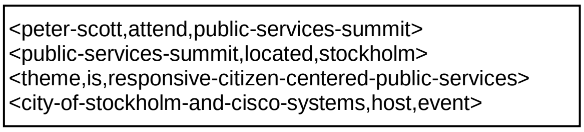
(1) “Enrico Motta, at KMi now, is leading a project on ….”.



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.
…

Figure 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>
```

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

Jape patterns 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.

Minipar’s representation is converted into a triple format and therefore the intermediate representation provided by both GATE and Minipar consists of triples of the type: <noun_phrase, verbal_expression, noun_phrase>.

3.3 Identifying 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 Figure 1. As first step we try to map the linguistic triple into an ontology triple, by using an adaptation of Aqualog’s Relation Similarity Service (RSS).

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 matching 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. Different strategies are employed to identify a matching for terms and relations, as we describe below.

Since we do not consider any interaction with the user in order to achieve a fully automated annotation process, other modules were developed to complete the mapping process even if there is no matching (Section 3.4) or if there is ambiguity (Section 3.5), according to RSS.

Strategies for mapping terms

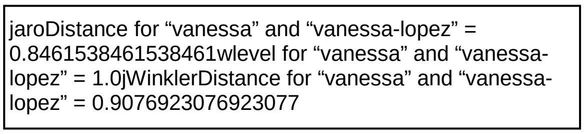
To map terms into entities, the following attempts are accomplished (in the given order):

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

2) Apply string similarity metrics[4] 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. Different combinations of threshold values for the metrics are considered. The elements in the 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.

For example, the similarity values returned for the term “vanessa” with instances potentially relevant for the mapping are given in Figure 4. The combination of thresholds is met for the instance “Vanessa Lopez”, and thus the mapping is accomplished. If there is still more than one possible mapping, we assume there is not enough evidence to map that term and discard the triple.



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

Figure 4. String similarity measures for the term “vanessa” and the instance “Vanessa Lopez”

[4] http://sourceforge.net/projects/simmetrics/

Strategies for 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 in the next section). 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 automatically created according to users' choices in the question answering system Aqualog. This lexicon contains ontology triples along with the original verbal expression, as illustrated in Table 1. The use of this lexicon represents a simplified form of pattern matching in which only exact matching is considered.

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

Table 1. Examples of lexicon patterns

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).

The application of these strategies to map the linguistic triples into existent or new instances and relations is described in what follows.

Applying RSS to map entities and 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. strategies for mapping terms). 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. strategies for mapping relations). 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:

<instance$_1$, (conceptual_relation)$^+$, 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):

<instance$_1$, *na* , ?> or <?, *na*, instance$_2$> or
<?, *na*, ?>

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

<instance$_1$, ?, 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 (Zhu et al, 2005).

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. All types of entities correspond to generic classes of our domain ontology, including: person, organization, event, publication, location, project, research-area, technology, 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, RSS is triggered 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 the possible relations between the classes of the recognized entities (cf. strategies for mapping relations).

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 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 compared against a given triple using its classes (already provided by the linguistic component or by ESpotter++) in order to classify relations in that triple as *relevant* or *non-relevant*.

The classification model is based on the approach presented in (Stevenson, 2004). 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) 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.

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

Table 2. Examples of seed patterns

Likewise (Stevenson, 2004), 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 module, 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 choose one relation by using the word sense disambiguation (**WSD**) system SenseLearner (Mihalcea and Csomai, 2005).

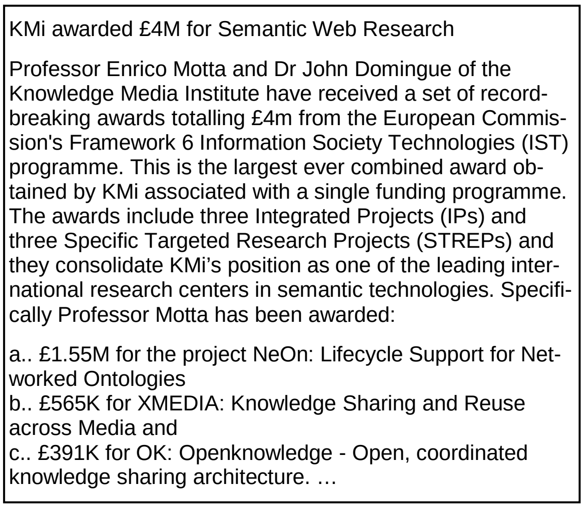
SenseLearner is supervised WSD system to disambiguate all open class words in any given text, after being trained on a small data set, according to global models for word categories. The current distribution includes two default models for verbs, which were trained on a corpus containing 200,000 content words of journalistic texts tagged with their WordNet senses. Since SenseLeaner requires a sense tagged corpus 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. 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 the words are annotated with POS tags. The use of lemmas of the words instead of the words yields better results, since the models were generated for lemmas. In our architecture, these 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. Since in WordNet synonyms usually represent different uses of the verb, the WSD module can identify in which sense the verb is being used in the sentence, allowing the system to choose one among all the matching options.

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) for the relation “head”. After getting all its synonyms 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 disambiguates the sense of “head” as “direct”.

3.6 Example of extracted relations

As an example of the relations that can be extracted in our approach, consider the representation of the entity “Enrico Motta” and all the relations involving this entity in Figure 5. The relations were extracted from the text in Figure 6.

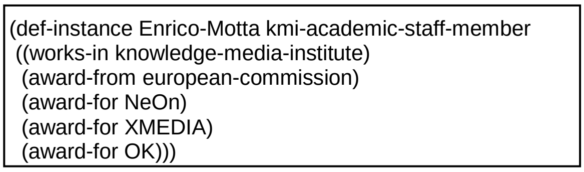


KMi awarded £4M for Semantic Web Research

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. …

Figure 5. 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)))
```

Figure 6. Semantic annotations produced for the news in Figure 5

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 a 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.

Acknowledgement

This research was supported by the Advanced Knowledge Technologies (AKT) project. AKT is an Interdisciplinary Research Collaboration (IRC), which is sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/N15764/01.

References

Massimiliano Ciaramita, Aldo Gangemi, Esther Ratsch, Jasmim Saric, Isabel Rojas. 2005. Unsupervised learning of semantic relations between concepts of a molecular biology ontology. *19th IJCAI*, pp. 659-664

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. *40th ACL Meeting*, Philadelphia.

Hamish Cunningham, Diana Maynard, and Valentin Tablan. 2000. JAPE: a Java Annotation Patterns Engine. *Tech. Report CS--00--10*, University of Sheffield, Department of Computer Science.

Christiane D. Fellbaum (ed). 1998. *Wordnet: An Electronic Lexical Database*. The MIT Press.

Pablo Gamallo, Marco Gonzalez, Alexandre Agustini, Gabriel Lopes, and Vera S. de Lima. 2002. Mapping syntactic dependencies onto semantic relations. *ECAI Workshop on Machine Learning and Natural Language Processing for Ontology Engineering*, Lyon, France.

Asuncion Gomez-Perez and David Manzano-Macho. 2003. *A Survey of Ontology Learning Methods and Techniques*. Deliverable 1.5, OntoWeb Project.

Jose Iria and Fabio Ciravegna. 2005. Relation Extraction for Mining the Semantic Web. *Dagstuhl Seminar on Machine Learning for the Semantic Web, Dagstuhl*, Germany.

Yuangui Lei, Marta Sabou, Vanessa Lopez, Jianhan Zhu, Victoria Uren, and Enrico Motta. 2006. An infrastructure for Acquiring High Quality Semantic Metadata. To appear in the *3rd ESWC*, Budva.

Dekang Lin. 1993. Principle based parsing without overgeneration. *31st ACL*, Columbus, pp. 112-120.

Vanessa Lopez, Michele Pasin, and Enrico Motta. 2005. AquaLog: An Ontology-portable Question Answering System for the Semantic Web. *2nd* E*SWC*, Creete, Grece.

Alexander D. Maedche. 2002. *Ontology Learning for the Semantic Web*, Kluwer Academic Publishers, Norwell, MA.

Scott Miller, Heidi Fox, Lance Ramshaw, and Ralph Weischedel. 2000. A novel use of statistical parsing to extract information from text. *6th ANLP-NAACL*, Seattle, pp. 226-233.

Rada Mihalcea and Andras Csomai. 2005. SenseLearner: Word Sense Disambiguation for All Words in Unrestricted Text. *43rd ACL Meeting*, Ann Arbor.

Marie-Laure Reinberger and Peter Spyns. 2004. Discovering knowledge in texts for the learning of DOGMA inspired ontologies. *ECAI 2004 Workshop on Ontology Learning and Population*, Valencia, pp. 19-24.

Dan Roth and Wen-tau Yih. 2002. Probabilistic reasoning for entity & relation recognition. *19th COLING*, Taipei, Taiwan, pp. 1-7.

Alexander Schutz and Paul Buitelaar. 2005. RelExt: A Tool for Relation Extraction from Text in Ontology Extension. *4th ISWC*, pp. 593-606.

Mark Stevenson. 2004. An Unsupervised WordNet-based Algorithm for Relation Extraction. *4th LREC Workshop Beyond Named Entity: Semantic Labeling for NLP Tasks*, Lisbon.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel Methods for Relation Extraction. *Journal of Machine Learning Research*, (3):1083-1106.

Shubin Zhao and Ralph Grishman. 2005. Extracting Relations with Integrated Information Using Kernel Methods. *43d ACL Meeting*, Ann Arbor.

Jianhan Zhu, Victoria Uren, and Enrico Motta. 2005. ESpotter: Adaptive Named Entity Recognition for Web Browsing. *3rd Conf. on Professional Knowledge Management*, Kaiserslautern, pp. 518-529.

Roman Yangarber, Ralph Grishman and Pasi Tapanainen, P. 2000. Unsupervised Discovery of Scenario-Level Patterns for Information Extraction. *6th ANLP*, pp. 282-289.

Author Index