Ontolex 2010

23rd International Conference on Computational Linguistics

Proceedings of the

6th Workshop on Ontologies and Lexical Resources

Alessandro Oltramari, Piek Vossen, and Qin Lu

22 August 2010
Beijing International Convention Center
Beijing, China

Produced by
Chinese Information Processing Society of China

To order the CD of Coling 2010 and its Workshop Proceedings, please contact:

Chinese Information Processing Society of China
No.4, Southern Fourth Street
Haidian District, Beijing, 100190
China
Tel: +86-010-62562916
Fax: +86-010-62562916
cips@iscas.ac.cn

Introduction

Welcome to the Coling Workshop on *Ontologies and Lexical Resources (OntoLex 2010)*.

As human linguistic practice reveals, accessing to concepts through natural language is the implicit pathway for enabling mutual comprehension and effective meaning negotiation between agents in a community. But, in order to exchange knowledge, we need to share the *conceptual models underlying the lexicon*, namely ontologies. These remarks become even more crucial when focusing on human-computer interaction. In this context, computational ontologies and human-language technologies converge in the task of providing the semantic description of knowledge contents (e.g. multimedia, web resources, services, etc.): underlying intended models need to be made explicit in order to become accessible by artificial agents and sharable with humans. According to this picture, 1) computational lexicons, whose aim is to make lexical-content machine-understandable, constitute a fundamental component to foster the (mono- and multi-linguistic) access to any knowledge content; 2) computational ontologies, on the other side, are necessary to capture the logical structure of those knowledge contents: both contribute to dig out the basic elements of a given semantic space (domain-dependent or general), characterizing the different relations holding among them.
In this general framework, the contributions presented under the scope of OntoLex 2010 (Ontologies and Lexical Resources) show in fact a variety of approaches under many respects. Some of the papers are oriented to describe the different construction processes of semantic resources (e.g., Daoud et al. and Nagata deal with two approaches based on Wikipedia), other papers are especially concerned with specific tasks and applications. Regarding the latter aspect, some contributions present proposals to enhance interoperability within the various standardization formats for linguistic and terminological descriptions (Peters, Vossen et al.) as well as exploiting specific algorithms for ontology matching. Some papers also focus on formal ontology, both at the level of theoretical analysis and at the level of specific categories and relations (see for example the paper by Bogulaslavsky). The investigated domains span from bio-surveillance (Conway et al.) through medicine; sentiment/opinion mining confirms to be an emergent area of interest too (see Cadilhac et al.). Automatic techniques and algorithms to extract terms and taxonomies are also introduced (Van der Plas, Nagata et al., vor der Brueck).
Originating in 2000, OntoLex is recognized as a common *meeting place* by a constantly growing interdisciplinary community of lexicographers, ontologists and computational linguists. Traditionally represented by researchers and practitioners from a variety of backgrounds (acquisition of lexical knowledge, ontology-based approaches to information extraction, ontology learning, ontology matching, etc.), OntoLex 2010's contributions confirm this trend in the Sixth edition of the workshop too, hosted by COLING conference for the first time. We think that the comprehensive perspective emerging from the 10 articles collected in these proceedings can help in progress towards next-generation knowledge systems based on the integration between ontologies and lexical resources.

July, 2010

Alessandro Oltramari, Laboratory for Applied Ontology (ISTC-CNR) & Department of Management and Engineering (University of Padua) (Italy)
Piek Vossen, Faculty of Arts, VU University Amsterdam (The Netherlands)
Qin Lu, Department of Computing, The Hong Kong Polytechnic University (HKSAR)

Workshop Organizers:

Alessandro Oltramari, Laboratory for Applied Ontology (ISTC-CNR) & Department of Management and Engineering (University of Padua) (Italy)
Piek Vossen, Faculty of Arts, VU University Amsterdam (The Netherlands)
Qin Lu, Department of Computing, The Hong Kong Polytechnic University (HKSAR)

Program Committee:

Christiane Fellbaum, Princeton University(U.S.A.)
Paul Buitelaar, DERI, National University of Ireland(Ireland)
Philipp Cimiano, Delft University of Technology(The Netherlands)
Emanuele Pianta, Fondazione Bruno Kessler(Italy)
Massimo Poesio, University of Trento(Italy)
Ed Hovy, University of Southern California(U.S.A.)
Bolette Pedersen,University of Copenhagen(Denmark)
John Bateman, University of Bremen(Germany)
Armando Stellato, Universit di Roma "Tor Vergata"(Italy)
Chu-Ren Huang, The Hong Kong Polytechnic University(HKSAR)
Guido Vetere, Center for Advanced Studies, IBM(Italy)
Laurent Prevot, University of Provence(France)
Kiril Simov,Bulgarian Academy of Sciences(Bulgaria)
Alessandro Lenci,University of Pisa(Italy)
Ernesto De Luca,Technische Universität Berlin(Germany)

Table of Contents

Conference Program

Sunday, August 22, 2010

8:30–9:30 Workshop registration

9:30–10:30 Keynote by Prof. Huang Chu-Ren Huang(HK Polytechnic University)

10:30–10:50 Coffee break

10:50–11:15 *KYOTO: an open platform for mining facts*
Piek Vossen, German Rigau, Eneko Agirre, Aitor Soroa, Monica Monachini and Roberto Bartolini

11:15–11:40 *Using Goi-Taikei as an Upper Ontology to Build a Large-Scale Japanese Ontology from Wikipedia*
Masaaki Nagata, Yumi Shibaki and Kazuhide Yamamoto

11:40–12:05 *Multilingual Lexical Network from the Archives of the Digital Silk Road*
Hans-Mohammad Daoud, Kyo Kageura, Christian Boitet, Asanobu Kitamoto and Mathieu Mangeot

12:05–12:30 *Finding Medical Term Variations using Parallel Corpora and Distributional Similarity*
Lonneke van der Plas and Jorg Tiedemann

12:30–14:30 Lunch break

14:30–14:55 *Learning Semantic Network Patterns for Hypernymy Extraction*
Tim vor der Bruck

14:55–15:20 *Intrinsic Property-based Taxonomic Relation Extraction from Category Structure*
DongHyun Choi, Eun-Kyung Kim, Sang-Ah Shim and Key-Sun Choi

15:20–15:45 *Developing a Biosurveillance Application Ontology for Influenza-Like-Illness*
Mike Conway, John Dowling and Wendy Chapman

15:45–16:00 Coffee break

16:00–16:25 *Interfacing the Lexicon and the Ontology in a Semantic Analyzer*
Igor Boguslavsky, Leonid Iomdin, Victor Sizov and Svetlana Timoshenko

Sunday, August 22, 2010 (continued)

16:25–16:40 Conclusions

Ontolexical resources for feature-based opinion mining: a case-study
Anais Cadilhac, Farah Benamara and Nathalie Aussenac-Gilles

KYOTO: an open platform for mining facts

Piek Vossen
VU University Amsterdam
p.vossen@let.vu.nl

German Rigau
Eneko Agirre
Aitor Soroa
University of the Basque Country
german.rigau/e.a-girre/a.soroa@ehu.es

Monica Monachini
Roberto Bartolini
Istituto di Linguistica Computazionale, CNR
monica.monachini/roberto.bartolini@ilc.cnr.it

Abstract

This document describes an open text-mining system that was developed for the Asian-European project KYOTO. The KYOTO system uses an open text representation format and a central ontology to enable extraction of knowledge and facts from large volumes of text in many different languages. We implemented a semantic tagging approach that performs off-line reasoning. Mining of facts and knowledge is achieved through a flexible pattern matching module that can work in much the same way for different languages, can handle efficiently large volumes of documents and is not restricted to a specific domain. We applied the system to an English database on estuaries.

1 Introduction

Traditionally, Information Extraction (IE) is the task of filling template information from previously unseen text which belongs to a predefined domain (Peshkin & Pfeffer 2003). Most systems in the Message Understanding Conferences (MUC, 1987-1998) and the Automatic Content Extraction program (ACE)[1] use a pipeline of tools to achieve this, ranging from sophisticated NLP tools (like deep parsing) to shallower text-processing (e.g. FASTUS (Appelt 1995)).

Standard IE systems are based on language-specific pattern matching (Kaiser & Miksch 2005), where each pattern consists of a regular expression and an associated mapping from syntactic to logical form. In general, the approaches can be categorized into two groups: (1) the Knowledge Engineering approach (Appelt et al.1995), and (2) the learning approach, such as AutoSlog (Appelt et al. 1993), SRV (Freitag 1998), or RAPIER (Califf & R. Mooney 1999). Another important system is GATE (Cunningham et al.2002), which is a platform for creating IE systems. It uses regular expressions, but it can also use ontologies to perform semantic inferences to constrain linguistic patterns semantically. The use of ontologies in IE is an emerging field (Bontcheva & Wilks 2004): linking text instances with elements belonging to the ontology, instead of consulting flat gazetteers.

The major disadvantage of traditional IE systems is that they focus on satisfying precise, narrow, pre-specified requests from small homogeneous corpora (e.g., extract information about terrorist events). Likewise, they are not flexible, are limited to specific types of knowledge and need to be built by knowledge engineers for each specific application and language. In fact most text mining systems are developed for a single domain and a single language, and are not able to handle knowledge expressed in different languages or expressed and conceptualized differently across cultures.

In this paper we describe an open platform for text-mining or IE that can be applied to many different languages in the same way using an open text representation system and a central on-

[1] http://www.itl.nist.gov/iad/mig//tests/ace

Proceedings of the 6th Workshop on Ontologies and Lexical Resources (Ontolex 2010), pages 1–10,
Beijing, August 2010

tology that is shared across languages. Ontological implications are inserted in the text through off-line reasoning and ontological tagging. The events and facts are extracted from large amounts of text using a flexible pattern-matching module, as specified by profiles which comprise ontological and shallow linguistic patterns. The system is developed in the Asian-European project KYOTO[2].

In the next section, we describe the general architecture of the KYOTO system. In section 3, we specify the knowledge structure that is used. Section 4, describes the off-line reasoning and ontological tagging. In section 5, we describe the module for mining knowledge from the text that is enriched with ontological statements. Finally in section 6, we describe the first results of applying the system to databases on Estuaries.

2 KYOTO overview

The KYOTO project allows communities to model terms and concepts in their domain and to use this knowledge to apply text mining on documents. The knowledge cycle in the KYOTO system starts with a set of source documents produced by the community, such as PDFs and websites. Linguistic processors apply tokenization, segmentation, morpho-syntactic analysis and semantic processing to the text in different languages. The semantic processing involves the detection of named-entities (persons, organizations, places, time-expressions) and determining the meaning of words in the text according to the given wordnet.

The output of the linguistic processors is stored in an XML annotation format that is the same for all the languages, called the KYOTO Annotation Format (KAF, Bosma et al 2009). This format incorporates standardized proposals for the linguistic annotation of text and represents them in an easy-to-use layered structure, which is compatible with the Linguistic Annotation Framework (LAF, Ide and Romary 2003). In KAF, words, terms, constituents and syntactic dependencies are stored in separate layers with references across the structures. This makes it easier to harmonize the output of linguistic processors for different languages and to add new semantic layers to the basic output, when needed (Bosma et al. 2009, Vossen et al. 2010). All modules in KYOTO draw their input from these structures. In fact, the word-sense disambiguation process is carried out to the same KAF annotation in different languages and is therefore the same for all the languages (Agirre et al. 2009). In the current system, there are processors for English, Dutch, Italian, Spanish, Basque, Chinese and Japanese.

The KYOTO system proceeds in 2 cycles (see Figure 1). In the 1st cycle, the **Tybot** (Term Yielding Robot) extracts the most relevant terms from the documents. The Tybot is another generic program that can do this for all the different languages in much the same way. The terms are organized as a structured hierarchy and, wherever possible, related to generic semantic databases, i.e. wordnets for each language. In the left part of Figure 1, we show those terms in the input document and their classification in wordnet. Terms in italics are present in the original wordnet, while underlined terms correspond to terms which were not in the original wordnet but were automatically discovered and linked to wordnet by Tybots. Straight terms correspond to hyperonyms in wordnet that do not necessarily occur in the text but are linked to ontological classes. The result of this 1st cycle is a domain wordnet for the target language.

The 2nd cycle of the system involves the actual extraction of factual knowledge from the documents by the **Kybots** (Knowledge Yielding Robots). Kybots use a collection of profiles that represent patterns of information of interest. In the profile, conceptual relations are expressed using ontological and morpho-syntactic linguistic patterns. Since the semantics is defined through the ontology, it is possible to detect similar data across documents in different languages, even if expressed differently. In Figure 1, we give an example of a conceptual pattern that relates organisms that live in habitats. The Kybot can combine morpho-syntactic and semantic patterns. When a match is detected, the instantiation of the pattern is saved in a formal representation, either in KAF or in RDF. Since the wordnets in different languages are mapped to the same ontology and the text in these languages is represented in the same KAF, similar patterns can easily be applied to multiple languages.

[2] Http://www.kyoto-project.eu

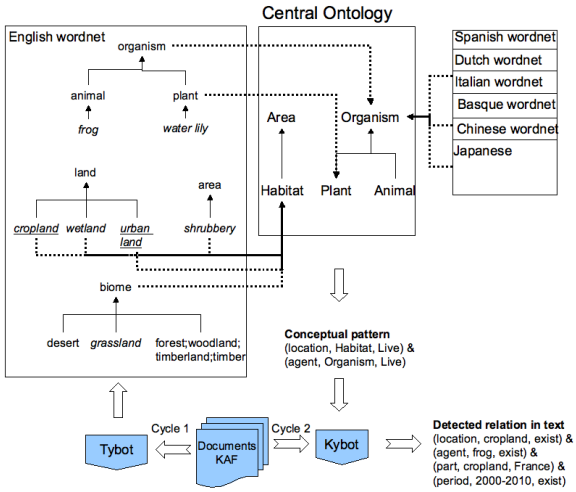

Figure 1: Two Cycles of processing in KYOTO

3 Ontological and lexical background knowledge

As a semantic background model, we defined a 3-layered knowledge architecture following the principle of the division of labour (Putnam 1975). In this model, the ontology does not need to be the central hub for all terms in a domain in all languages. Following the division of labour principle, we can state that a computer does not need to distinguish between instances of a European Tree Frog and a Glass Tree frog. We assume that rigid concepts (as defined by Guarino and Welty 2002) are known to the domain experts and do not need to be defined formally in the ontology but can remain in the available background resources, such as databases with millions of species. Terms in the documents are mostly non-rigid, e.g. *endangered frogs*, *invasive frogs*. Such non-rigid terms refer to instances of species in contextual circumstances. The processes and states are the important pieces of information that matter to the users and are useful for mining text. The model therefore distinguishes between background vocabularies, domain terms, wordnets and the central ontology. The background vocabularies are automatically aligned to wordnet, where we assume that hyponymy relations to rigid synsets in wordnet declare those subconcepts as rigid subtypes too, without the necessity to include them in the ontology. For non-rigid terms, we defined a set of mapping relations to the ontology through which we express their non-rigid involvement in these processes and states. Likewise, the ontology has been extended with processes and states for the domain and verbs and adjectives have been mapped to be able to detect expressions in text.

The 3-layered knowledge model combines the efforts from 3 different communities:

1. Domain experts in social communities that continuously build background vocabularies;
2. Wordnet specialists that define the basic semantic model for general concepts for a language
3. Semantic Web specialists that define top-level and domain-specific ontologies that capture formal definitions of concepts;

We formalized the relations between these repositories so that they can developed separately but combined within KYOTO to form a coherent and formal model.

3.1 Ontology

The KYOTO ontology currently consists of 1149 classes divided over three layers. The top layer is based on DOLCE (DOLCE-Lite-Plus version 3.9.7, Masolo et al 2003) and OntoWordNet. This layer of the ontology has been modified for our purposes (Herold et. al. 2009). The second layer consists of so-called Base Concepts (BCs) derived from various wordnets (Vossen 1998, Izquierdo et al. 2007). Examples of BCs are: *building, vehicle, animal, plant, change, move, size, weight*. The BCs are those synsets in WordNet 3.0 that have the most relations with other synsets in the wordnet hierarchies and are selected in a way that ensures complete coverage of the nominal and verbal part of WordNet. This has been completed for the nouns (about 500 synsets). The ontology has also been adapted to include important concepts in the domain. Special attention has been paid to represents the processes (**perdurants**) in which objects (**endurants**) of the domain are involved and qualities they may have. This is typically the information that is found in documents on the environment. We thus added 40 new event classes for representing important verbs (e.g. *pollute*, *absorb*, *damage*, *drain*) and 115 new qualities and quality-regions for representing important adjectives (e.g. *airborne*, *acid*, (*un*)*healthy*, *clear*). The full

ontology can be downloaded from the KYOTO website, free for use. A considerable set of general verbs and adjectives (relevant for for the domain) have then been mapped to ontological classes: 189 verbal synsets and 222 adjectival synsets.

The 500 nominal BCs are connected to the complete WordNet hierarchy, whereas the 189 verbs represent 5,978 more specific verbal synsets and the 222 adjectives represent 1,081 adjectival synsets through the wordnet relations.

This basic ontology and the mapping to WordNet are used to model the shared and language-neutral concepts and relations in the domain. Instances are excluded from the ontology. Instances will be detected in the documents and will be mapped to the ontology through instance to ontology relations (see below). Likewise, we make a clear separation between the ontological model and the instantiation of the model as described in the text.

3.2 Wordnet to ontology mappings

In addition to the ontology, we have wordnets for each language in the domain. In addition to the regular synset to synset relations in the wordnet, we will have a specific set of relations for mapping the synsets to the ontology, which are all prefixed with *sc_* standing for synset-to-concept. We differentiate between rigid and non-rigid concepts in the wordnets through the mapping relations:

- sc_**equivalenceOf**: the synset is fully equivalent to the ontology Type & inherits all properties; the synset is Rigid
- sc_ **subclassOf**: the synset is a proper subclass of the ontology Type & inherits all properties; the synset is Rigid
- sc_**domainOf**: the synset is not a proper subclass of the ontology Type & is not disjoint (therefore orthogonal) with other synsets that are mapped to the same Type either through sc_subclassOf or sc_domainOf; the synset is non-Rigid but still inherits all properties of the target ontology Type; the synset is also related to a Role with a sc_playRole relation
- sc_**playRole**: the synset denotes instances for which the context of the Role applies for some period of time but this is not essential for the existence of the instances, i.e. if the context ceases to exist then the instances may still exist (Mizoguchi et al. 2007).[3]
- sc_**participantOf**: instances of the concept (denoted by the synset) participate in some endurant, where the specific role relation is indicated by the playRole mapping.
- sc_**hasState**: instances of the concept are in a particular state which is not essential and can be changed. There is no need to represent the role for a stative perdurant.

This model extends existing WordNet to ontology mappings. For instance, in the SUMO to Wordnet mapping (Niles and Pease 2003), only the *sc_equivalenceOf* and *sc_subclassOf* relations are used, represented by the symbols '=' and '+' respectively. The SUMO-Wordnet mapping likewise does not systematically distinguish rigid from non-rigid synsets. In our model, we separate the linguistically and culturally specific vocabularies from the shared ontology while using the ontology to interface the concepts used by the various communities.

Using these mapping relations, we can express that the synset for *duck* (which has a hypernym relation to the synset *bird*, which, in its turn, has an equivalence relation to the ontology class *bird*) is thus a proper subclassOf the ontology class *bird*:

wn:duck hypernym *wn:bird*
wn:bird sc_equivalenceOf *ont:bird*

For a concept such as *migratory bird*, which is also a hyponym of *bird* in wordnet but not a proper subclass as a non-rigid concept, we thus create the following mapping:

wn:migratory bird
→ sc_domainOf *ont:bird*
→ sc_playRole *ont:done-by*
→ sc_participantOf *ont:migration*

This mapping indicates that the synset is used to refer to instances of endurants (not subclasses!), where the domain is restricted to birds. Furthermore, these instances participate in the process of

[3] Some terms involve more than one role, e.g. gas-powered-vehicle. Secondary participants are related through sc_**hasCoParticipant** and sc_**playCoRole** mappings.

migration in the role of *done-by*. The properties of the process migration are further defined in the ontology, which indicates that it is a active-change-of-location done-by some endurant, going from a source, via a path to some destination. The mapping relations from the wordnet to the ontology, need to satisfy the constraints of the ontology, i.e. only roles can be expressed that are compatible with the role-schema of the process in which they participate.

For implied non-essential states, we use the *sc_hasState* relation to express that a synset such as *wild dog* refers to instances of dogs that life in the *wild* but can stop being *wild*:

wn:wild dog → sc_domainOf *ont:dog*
wn:wild dog → sc_hasState *ont:wild*

Ideally, all processes and states that can be applied to endurants should be defined in the ontology. This may hold for most verbs and adjectives in languages, which do not tend to extend in specific domains and are part of the general vocabulary (e.g. *to pollute, to reduce, wild*). However, domain specific text contain many new nominal terms that refer to domain-specific processes and states, e.g. *air pollution, nitrogen pollution, nitrogen reduction*. These terms are equally relevant as their counter-parts that refer to endurants involved in similar processes, e.g. *polluted air, polluting nitrogen or reduced nitrogen*. We therefore use the reverse participant and role mappings to be able to define such terms for processes as subclasses of more general processes involving specific participants in a specified role:

wn:air pollution
→ sc_subcassOf *ont:pollution (perdurant)*
→ sc_hasParticipant *ont:air*
→ sc_hasRole *ont:patient*
wn:nitrogen pollution
→ sc_subcassOf *ont:pollution (perdurant)*
→ sc_hasParticipant *ont:nitrogen*
→ sc_hasRole *ont:done-by*

Further mapping relations are described in the documentation on the KYOTO website. Through the mapping relations, we can keep the ontology relatively small and compact whereas we can still define the richness of the vocabularies of languages in a precise way. The classes in the ontology can be defined using rich axioms that model precise implications for inferencing. The wordnet to synset mappings can be used to define rather basic relations relative to the given ontology that still captures the semantics of the terms. The term definitions capture both relevance and perspective (those relations that matter from the point of the view of the term), on the one hand, and some semantics with respect to the concepts that are involved and their (role) relation on the other hand. Likewise, the KYOTO system can model the linguistic and cultural diversity of languages in a domain but at the same time keep a firm anchoring to a basic and compact ontology.

3.3 Domain wordnet

We selected 3 representative documents on estuaries to extract relevant terms for the domain using the Tybot module. The terms have been related through structural relations, e.g. *nitrogen pollution* is a hyponym of *pollution*, and through WordNet synsets that are assigned through WSD of the text. We extracted 3950 candidate terms form the KAF representations of the documents. Most of these are nouns (2818 terms). The nominal terms matched for 40% with wordnet synsets, the verbs and adjectives for 98% and 85% respectively. For the domain wordnet, we restricted ourselves to the nouns. From the new nominal terms, environmentalists selected 390 terms that they deem to be important. These terms are connected to parent terms, which ultimately are connected to wordnet synsets. The final domain wordnet contains 659 synsets: 197 synsets from the generic wordnet and 462 new synsets connected to the former. The domain wordnet synsets got 990 mappings to the ontology, using the relations described in the previous section. There are 86 synsets that have a sc_domainOf mapping, indicating that they are non-rigid. Note that hyponyms of these synsets are also non-rigid by definition. These non-rigid synsets have complex mappings to processes and states in which they are involved. The domain wordnet can be downloaded from the KYOTO website, free for use.

```
<term lemma="pollution" pos="N" tid="t13444" type="open">
  <externalReferences>
   <externalRef reference="eng-30-00191142-n" reftype="baseConcept" resource="wn30g"/>
   <externalRef reference="Kyoto#change-eng-3.0-00191142-n" reftype="sc_subClassOf" resource="ontology">
      <externalRef reftype="SubClassOf" reference="DOLCE-Lite.owl#contamination_pollution"/>
      <externalRef reftype="SubClassOf" reference="DOLCE-Lite.owl#accomplishment" status="implied"/>
      <externalRef reftype="SubClassOf" reference="DOLCE-Lite.owl#event" status="implied"/>
      <externalRef reftype="SubClassOf" reference="DOLCE-Lite.owl#perdurant" status="implied"/>
      <externalRef reftype="DOLCE-Lite.owl#part" reference="DOLCE-Lite.owl#perdurant" status="implied"/>
      <externalRef reftype="DOLCE-Lite.owl#specific-constant-constituent" reference="DOLCE-Lite.owl#perdurant"
status="implied"/>
      <externalRef reftype="DOLCE-Lite.owl#has-quality" reference="DOLCE-Lite.owl#temporal-quality" status="implied"/>
      <externalRef reftype="SubClassOf" reference="DOLCE-Lite.owl#spatio-temporal-particular" status="implied"/>
      <externalRef reftype="DOLCE-Lite.owl#participant" reference="DOLCE-Lite.owl#endurant" status="implied"/>
      <externalRef reftype="DOLCE-Lite.owl#has-quality" reference="DOLCE-Lite.owl#temporal-location_q" status="im-
plied"/>
    <externalRef reftype="SubClassOf" reference="DOLCE-Lite.owl#particular" status="implied"/>
    </externalRef>
  </externalReferences>
</term>
```

Figure 2: An example of an OntoTagged output

```
<kprofile>
 <variables>
  <var name="x" type="term" pos="N"/>
  <var name="y" type="term"
       lemma="produce | generate | release | ! create"/>
  <var name="z" type="term"
       reference="DOLCE-Lite.owl#contamination_pollution"
       reftype="SubClassOf"/>
 </variables>
 <relations>
  <root span="y"/>
  <rel span="x" pivot="y" direction="preceding"/>
  <rel span="z" pivot="y" direction="following"/>
 </relations>
 <events>
  <event target="$y/@tid" lemma="$y/@lemma" pos="$y/@pos"/>
  <role target="$x/@tid" rtype="agent" lemma="$x/@lemma"/>
  <role target="$z/@tid" rtype="patient"lemma="$z/@lemma"/>$
 </events>
</kprofile>
```

Figure 3: An example of a Kybot profile

```
<kybotOut>
 <doc name="11767.mw.wsd.ne.onto.kaf">
  <event eid="e1" lemma="generate" pos="V" target="t3504"/>
  <role rid="r1" lemma="industry" rtype="agent" target="t3493" pos="N" event="e1"/>
  <role rid="r2" lemma="pollution" rtype="patient" target="t3495" pos="N" event="e1"/>
 </doc>
 <doc name="16266.mw.wsd.ne.onto.kaf">
  <event eid="e2" lemma="release" pos="V" target="t97"/>
  <role rid="r3" lemma="fuel" rtype="agent" target="t96" pos="N" event="e2"/>
  <role rid="r4" lemma="exhaust_gas" rtype="patient" target="t101" pos="V" event="e2"/>
 </doc>
</kybotOut>
```

Figure 4: An example of a Kybot output

4 Off-line reasoning and ontological tagging

The ontological tagging represents the last phase in the KYOTO Linguistic Processor annotation pipeline. It consists of a three-step module devised to enrich the KAF documents with knowledge derived from the ontology. For each synset connected to a term, the first step adds the Base Concepts to which the synset is related through the wordnet taxonomical relations. Then, through the synset to ontology mapping, it adds the corresponding ontology type with appropriate relations. Once each synset is specified as to its ontology type, the last ontotagging step inserts the full set of ontological implications that follow from the explicit ontology. The explicit ontology is a new data structure consisting of a table with all ontology nodes and all ontological implications expressed. The main purpose is to optimize

the performance of the mining module over large quantities of documents. The advantage for Kybots from ontotagging are many. First of all, they are able to run and apply pattern-matching to Base Concepts and ontological classes rather than just to words or synsets. Moreover, by making explicit the implicit ontological statements, Kybots are able to find the same relations hidden in different expressions with different surface realizations: *fish migration, migratory fish, migration of fish, fishes that migrate,* that directly or indirectly express the same relations. With ontotagging, they share the same ontological implications which will allow Kybots to apply the same patterns and perform the extraction of facts. The implications will be represented in the same way across different languages, thus facilitating cross-lingual extraction of facts. Lastly, ontotagging is a kind of off-line ontological reasoning: without doing reasoning over concepts, Kybots substantially improve their performance. Figure 2 shows the result of onto-tagging for the term *pollution*.

5 Event and fact extraction

Kybots (Knowledge Yielding Robots) are computer programs that use the mined concepts and the generic concepts already connected to the language wordnets and the KYOTO ontology to extract actual concept instances and relations in KAF documents. Kybots incorporate technology for the extraction of relationships, either eventual or not, relative to the general or domain concepts already captured by the Tybots. That is, the extraction of factual knowledge is being carried out by the Kybot server by processing Kybot profiles on the linguistically enriched documents.

Kybots are defined following a declarative format, the so called *Kybot profiles*, which describe general morpho-syntactic and semantic conditions on sequences of terms. Profiles are compiled to generate the Kybots, which scan over KAF documents searching for the patterns and extract the relevant information from each matching.

Linguistic patterns include morphologic constraints and also semantic conditions the matched terms must hold. Kybot are thus able to search for term lemmas or part-of-speech tags but also for terms linked to ontological process and states using the mappings described in Section 3.2. Thus, it is possible to detect similar eventual information across documents in different languages, even if expressed differently.

5.1 Example of a Kybot Profile

Kybot Profiles are described using XML syntax. Figure 3 presents an example of a profile. Kybot profiles consist of three main parts:

•*Variable declaration* (`<variables>` element): In this section the search entities are defined. The example defines three variables: **`x`** (denoting terms whose part-of-speech is noun), **`y`** (which are terms whose lemma is “release”, “produce” or “generate” but not “create”) and **`z`** (terms linked to the ontological endurant “DOLCE-Lite.owl#contamination_pollution”, meaning ``being contaminated with harmful substances").

•*Declarations of the relations among variables* (`<rel>` element): specify the relations among the previously defined variables. The example profile specifies **`y`** as the main pivot, and states that variable **`x`** must be preceding variable **`y`** in the same sentence, and that variable **`z`** must be following variable **`y`**. Thus, the Kybot will search for patterns like '**`x`** → **`y`** → **`z`**' in a sentence.

•*Output template* (`<events>` element): describes the output to be produced on every matching. In the example, each match generates a new event targeting term **`y`**, which becomes the main term of the event. It also fills two roles of the event, the 'agent' role filled by term **`x`** and 'patient' role, filled by **`z`**.

Figure 4 presents the output of the Kybot when applied against the benchmark documents. The Kybot output follows the stand-off architecture when producing new information, and it thus forms a new KAF layer on the original documents.

6 Experimental results

We applied the KYOTO system and resources to English documents on estuaries. We collected 50 URLs for two English estuaries: the Humber Estuary in Hull (UK) and the Chesapeake Bay estuary in the US and for background documents on bird migration, sedimentation, habitat destruction, and climate change. In addition to the webpages, we extracted 815 PDF files from the sites. In total, 4625 files have been extracted. All

the documents have been processed by the linguistic processor for English, which generated KAF representations for all the documents. From this database, 3 documents were selected for benchmarking.

The documents were processed by applying multiword tagging, word-sense-disambiguation, named-entity-recognition and the ontological tagging to the 3 documents and to the complete database; This was done twice: once without the domain model and once with the domain model. We thus created 4 datasets: 3 benchmark documents processed with and without the domain model; the complete database processed with and without the domain model.

Furthermore, we created Kybot profiles based on the type of information represented in the domain model. We applied the Kybots to all 4 data sets. We generate the following data files through an WN-LMF export of the domain wordnet:

1. a set of domain multiwords for the multiword tagger
2. an extension of the lexicon and the graph of concepts that is used by the WSD module
3. an extension of the wordnet-to-ontology mappings for the ontotagger

In addition, we constructed mapping lists for all WordNet 3.0 synsets to Base Concepts and to adjective and verbs that are matched to the ontology. These mappings provide the generic conceptual model based on wordnet and on the ontology.

Table 1 shows the effects of using the domain model for the first 3 modules. We can see that the domain model has a clear effect on the multiword detection in the 3 evaluation documents. Using the domain model, 600 multiwords have been detected, against 145 with just the generic wordnet. This is obvious since the terms are extracted from the same documents. However, when applying it to the complete database, we see that still over 2,300 more multiwords have been detected using the domain wordnet. Note that the domain wordnet has only 97 multiwords and the generic wordnet has 19,126 multiwords. So 0.5% of the multiwords in the domain wordnet add 1.5 times more multiword tokens in the database. The third row specifies the number of synsets that have been assigned. We can see that for the domain model almost 400 more synsets have been detected. In the case of the full estuary database, we see that relatively few more have been detected, almost 1,500 while the database is 80 times as big. If we look more closely at the numbers of actual domain synsets detected, we see the following results. In the benchmark documents 637 (or 5%) of the synsets is a domain wordnet synset, whereas 5,353 synsets are domain synsets in the full estuary database, which is only 0.52%. Note that in KAF multiwords are represented both as a single terms and in terms of their elements. The WSD module assigns synsets to both. The domain model can thus only add synsets compared to the processing without the domain.

Finally, if we look at the named-entity-recognition module, we see a slight negative effect for the detection of named-entities due to the domain model. The named-entity-recognition module does not consider the elements of multiwords but just the multiword terms as a whole. Grouping terms as multiwords thus leads to less named-entities being detected. This is not necessarily a bad things, since the detection heavily over-generates and could have now more precision.

	bench mark documents (3)		estuary documents (4742)	
	No Domain	Domain	No Domain	Domain
terms	22,204	22,204	2,419,839	2,419,839
multiwords	145	600	4,389	6,671
synsets	12,526	12,910	1,021,598	1,023,017
ne location	158	126	41,681	40,714
ne date	67	66	10,288	10,233

Table 1: Statistics on processing the estuary documents with and without domain model

	bench mark documents (3)				estuary documents (4272	
	No Domain		Domain		Domain	
ontology references	555,677		576,432		48,708,300	
implied ontology referer	457,332	82.30%	474,916	82.39%	40,523,452	83.20%
direct ontology referenc	53,178	9.57%	54,769	9.50%	4,377,814	8.99%
domain synset to ontolo	45,167	8.13%	46,747	8.11%	3,807,034	7.82%

Table 2: Ontological implications for the four data sets

Table 2 shows the effect of inserting ontological implications into the text representation.For the benchmark documents, we see that more than half a million ontological implications have been inserted. Of these, 82% are implied references, that are extracted from the explicit ontology on the basis of a direct mapping to the ontology. About 8% of the mappings are synset-to-ontology mappings (sc) and 9.5% are mappings representing the subclass hierarchy. The differences between using the domain model and not-using the domain model are minimal. For the complete database, the implications are 80 times as much but the proportions are similar.

Table 3 shows the type of sc-relations that occur. Obviously, sc_subClassOf and sc_equivalentOf are the most frequent. Nevertheless, we still find about 500 mappings that present the participation in a process or state.

```
   30  reftype="sc_playCoRole"
   32  reftype="sc_hasCoParticipant"
   42  reftype="sc_partOf"
   59  reftype="sc_stateOf"
   92  reftype="sc_playRole"
   94  reftype="sc_hasRole"
   97  reftype="sc_participantOf"
  105  reftype="sc_hasParticipant"
  128  reftype="sc_domainOf"
  169  reftype="sc_hasState"
  312  reftype="sc_hasPart"
 3637  reftype="sc_equivalentOf"
42048  reftype="sc_subClassOf"
```

Table 3: Type of relations for the wordnet to ontology mappings using the domain model

The table clearly shows the impact of role relations that are encoded in the domain wordnet. When we extract the mappings for the files without the domain model (ony using the mappings to the generic wordnet), we get only equivalence and subclass mappings.

Finally to complete the knowledge cycle, we created a few Kybot profiles for extracting events from the onto-tagged documents. As an initial test, 3 profiles have been created:

1. events of destruction
2. destructions of locations
3. destruction of objects

Using these profiles, we extracted 211 events from the 3 benchmark documents with 396 roles. The profiles are created to run over the ontological types inserted by the ontotagger, e.g. restricted to events and change_of_integrity. Despite the generality of the profiles, we still see a clear signature of the domain in the output. This is a good indication that we will be able to extract valuable events from the data, even though the ontotagger generates a massive amount of implications. Especially events that combine multiple roles appear to give rich information. For example, the following sentence:

"One of the greatest challenges to restoration is continued population growth and development, which destroys forests, wetlands and other natural areas"

yielded the following output:

```
  <event target="t1471" lemma="destroy" pos="V"
eid="e74"/>
  <role target="t1477" rtype="patient" lemma="area"
pos="N" event="e74" rid="r138"/>
  <role target="t1472" rtype="patient"
lemma="forest" pos="N" event="e74" rid="r151"/>
  <role target="t1469" rtype="actor" lemma="devel-
opment" pos="N" event="e74" rid="r180"/>
```

Running the full set of profiles on the complete database with almost 60 million ontological statements took about 2 hours. This shows that our approach is scalable and efficient.

7 Conclusions

In this paper, we described an open platform for text-mining using wordnets and a central ontology. The system can be used across different languages and can be tailored to mine any type of conceptual relations. It can handle semantic implications that are expressed in very different linguistic expressions and yield systematic output. As future work, we will carry out benchmarking and testing of the mining of events, both for English and for the other languages in the KYOTO project.

Acknowledgements

The KYOTO project is co-funded by EU - FP7 ICT Work Programme 2007 under Challenge 4 - Digital libraries and Content, Objective ICT-2007.4.2 (ICT-2007.4.4): Intelligent Contsent and Semantics (challenge 4.2). The Asian partners from Tapei and Kyoto are funded from national funds. This work has been also supported by Spanish project KNOW-2 (TIN2009-14715-C04-01).

References

Agirre, E., & Soroa, A. (2009) Personalizing PageRank for Word Sense Disambiguation. Proceedings of the 12th EACL, 2009. Athens, Greece.

Agirre, E., Lopez de Lacalle, O., & Soroa, A. (2009) Knowledge-based WSD and specific domains: performing over supervised WSD. Proceedings of IJCAI. Pasadena, USA. http://ixa.si.ehu.es/ukb

Álvez J., Atserias J., Carrera J., Climent S., Laparra E., Oliver A. and Rigau G. (2008) Complete and Consistent Annotation of WordNet using the Top Concept Ontology. Proceedings of LREC'08, Marrakesh, Morroco. 2008.

Appelt Douglas E., Jerry R. Hobbs, John Bear, David Israel, Megumi Kameyama, Andrew Kehler, David Martin, Karen Myers and Mabry Tyson. Description of the FASTUS System Used for MUC-6. In Proceedings of MUC-6, pages 237–248. San Mateo, Morgan Kaufmann, 1995.

Auer A., C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak and Z. Ives. DBpedia: A Nucleus for a Web of Open Data. In Proceedings of the International Semantic Web Conference (ISWC), volume 4825 of Lecture Notes in Computer Science, pages 722-735. 2007.

Bosma, W., Vossen, P., Soroa, A. , Rigau, G., Tesconi, M., Marchetti, A., Monachini, M., & Apiprandi, C. (2009) KAF: a generic semantic annotation format. In Proceedings of the 5th International Conference on Generative Approaches to the Lexicon Sept 17-19, 2009, Pisa, Italy.

Fellbaum, C. (Ed.) (1998) WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press.

Freitag, D. (1998) Information extraction from html: Application of a general machine learning approach. In Proceedings of the Fifteenth National Conference on Artificial Intelligence, 1998.

Gangemi A., Guarino N., Masolo C., Oltramari A., Schneider L. (2002) Sweetening Ontologies with DOLCE. Proceedings of EKAW. 2002

Ide, N. and L. Romary. 2003. Outline of the inter- national standard Linguistic Annotation Framework. In *Proceedings of ACL 2003 Workshop on Linguistic Annotation: Getting the Model Right*, pages 1–5.

Izquierdo R., Suárez A. & Rigau G. Exploring the Automatic Selection of Basic Level Concepts. Proceedings of RANLP'07, Borovetz, Bulgaria. September, 2007.

Masolo, C., Borgo, S., Gangemi, A., Guarino, N. & Oltramari, A. (2003) WonderWeb Deliverable D18: Ontology Library, ISTC-CNR, Trento, Italy.

Mizoguchi R., Sunagawa E., Kozaki K. & Kitamura Y. (2007 A Model of Roles within an Ontology Development Tool: Hozo. Journal of Applied Ontology, Vol.2, No.2, 159-179.

Niles, I. & Pease, A. (2001) Formal Ontology in Information Systems. Proceedings of the international Conference on Formal Ontology in Information Systems – Vol. 2001 Ogunquit, Maine, USA

Niles, I. and A. Pease. Linking lexicons and ontologies: Mapping WordNet to the Suggested Upper Merged Ontology. In Proc. IEEE IKE, pages 412–416, 2003.

Vossen, P. (Ed.) (1998) EuroWordNet: a multilingual database with lexical semantic networks for European Languages. Kluwer, Dordrecht.

Vossen P., W. Bosma, E. Agirre, G. Rigau, A. Soroa (2010) A full Knowledge Cycle for Semantic Interoperability. Proceedings of the 5th Joint ISO-ACL/SIGSEM Workshop on Interoperable Semantic Annotation, (ICGL 2010) Hong Kong, 2010.

Using Goi-Taikei as an Upper Ontology to Build a Large-Scale Japanese Ontology from Wikipedia

Masaaki Nagata
NTT Communication Science Laboratories
nagata.masaaki@labs.ntt.co.jp

Yumi Shibaki and **Kazuhide Yamamoto**
Nagaoka University of Technology
{shibaki,yamamoto}@jnlp.org

Abstract

We present a novel method for building a large-scale Japanese ontology from Wikipedia using one of the largest Japanese thesauri, *Nihongo Goi-Taikei* (referred to hereafter as "Goi-Taikei") as an upper ontology. First, The leaf categories in the Goi-Taikei hierarchy are semi-automatically aligned with semantically equivalent Wikipedia categories. Then, their subcategories are created automatically by detecting *is-a* links in the Wikipedia category network below the junction using the knowledge defined in Goi-Taikei above the junction. The resulting ontology has a well-defined taxonomy in the upper level and a fine-grained taxonomy in the lower level with a large number of up-to-date instances. A sample evaluation shows that the precisions of the extracted categories and instances are 92.8% and 98.6%, respectively.

1 Introduction

In recent years, we have become increasingly aware of the need for up-to-date knowledge bases offering broad-coverage in order to implement practical semantic inference engines for advanced applications such as question answering, summarization and textual entailment recognition. One promising approach involves automatically extracting a large comprehensive ontology from Wikipedia, a freely available online encyclopedia with a wide variety of information. One problem with previous such efforts is that the resulting ontology is either fragmentary or trivial.

Ponzetto and Strube (2007) presents a set of lightweight heuristics such as *head matching* and *modifier matching* for distinguishing between *is-a* and *not-is-a* links in the Wikipedia category network. The most powerful heuristics is head matching in which a category link is labeled as *is-a* if the two categories share the same head lemma, such as CAPITALS IN ASIA and CAPITALS. For Japanese, Sakurai et al. (2008) present a method equivalent to head matching in Japanese. As Japanese is a head final language, they introduced a heuristics called *suffix matching* in which a category link is labeled as *is-a* if one category is the suffix of the other category, such as 日本の空港 (airports in Japan) and 空港 (airports). The problem with the ontology extracted by these two methods is that it is not a single interconnected taxonomy, but a set of taxonomic trees.

One way to make a single taxonomy is to use an existing large-scale taxonomy as a core for the resulting ontology. In YAGO, Suchanek et al. (2007) merged English WordNet and Wikipedia by adding instances (namely Wikipedia articles) to the *is-a* hierarchy of WordNet. Of the categories assigned to a Wikipedia article, they regarded one with a plural head noun as the article's hypernym, which is called a *conceptual category*. They then linked the conceptual category to a WordNet synset by heuristic rules including head matching. For Japanese, Kobayashi et al. (2008) present an attempt equivalent to YAGO, where they merged Goi-Taikei and Japanese Wikipedia. The problem with these two methods is that the core taxonomy is extended only one level although many new instances are added. They cannot make the most of the fine-grained taxonomic

Proceedings of the 6th Workshop on Ontologies and Lexical Resources (Ontolex 2010), pages 11–18,
Beijing, August 2010

information contained in the Wikipedia category network.

In this paper, we present a novel method for building a single interconnected ontology from Wikipedia, with a fine-grained taxonomy in the lower level, by using a manually constructed thesaurus as its upper ontology. In the following sections, we first describe the language resources used in this work. We then describe a semi-automatic method for building the ontology and report our experimental results.

2 Language Resources

2.1 *Nihongo Goi-Taikei*

Nihongo Goi-Taikei (日本語語彙大系, 'comprehensive outline of Japanese vocabulary')[1] is one of the largest and best known Japanese thesauri (Ikehara et al., 1997). It was originally developed as a dictionary for a Japanese-to-English machine translation system in the early 90's. It was then published as a book in 5 volumes in 1997 and as a CD-ROM in 1999. It contains about 300,000 Japanese words and the meanings of each word are described by using 2,715 hierarchical semantic categories. Each word has up to 5 semantic categories in order of frequency in use, and each category is assigned with a unique ID number and category name such as 4:person and 388:place[2].

Goi-Taikei has different semantic category hierarchies for common nouns, proper nouns, and verbs, respectively. We used only the common noun category in this work. For simplicity, we mapped all proper nouns in the proper noun category to the equivalent common noun category using the category mapping table shown in the Goi-Taikei book.

Figure 1 shows the top three layers for common nouns[3]. For example, the transliterated Japanese word *raita* (ライター) has two semantic categories 353:author and 915:household appliance. The former originates with the English word "writer" while the latter originates with English word "lighter". By climbing up the Goi-Taikei category hierarchy, we can infer that the former refers to a human being (4:person) while the latter refers to a physical object (533:concrete object).

2.2 Japanese Wikipedia

Wikipedia is a free, multilingual, on-line encyclopedia actively developed by a large number of volunteers. Japanese Wikipedia now has about 500,000 articles. Figure 2 shows examples of an article page and a category page. An article page has a title, body, and categories. In most articles, the first sentence of the body gives the definition of the title. A category also has a title, body, and categories. Its title is prefixed with "Category:" and its body includes a list of articles that belong to the category.

Although the Wikipedia category system is organized in a hierarchal manner, it is not a taxonomy but a thematic classification. An article could belong to many categories and the category network has loops. The relations between linked categories are chaotic, but the lower the category link is in the hierarchy, the more it is likely to be an *is-a* relation. For example, the category link between カクテル (COCKTAIL) and 酒 (ALCOHOLIC BEVERAGE) is an *is-a* relation. Although the article シェイカー (shaker) is in the category カクテル (COCKTAIL), a shaker is not a cocktail but an appliance. Extracting a taxonomy from the Wikipedia category network is not trivial.

3 Ontology Building Method

Figure 3 shows an outline of the proposed ontology building method. We first semi-automatically align each leaf category in the Goi-Taikei category hierarchy with one or more Wikipedia categories. We call a Wikipedia category aligned with a Goi-Taikei category a *junction category*. We then extend each Goi-Taikei leaf category by detecting the *is-a* links below the junction category in the Wikipedia category network using the knowledge defined above the junction category in Goi-Taikei .

[1] Referred to as "Goi-Taikei" unless otherwise noted.

[2] We use Sans Serif for the Goi-Taikei category and SMALL CAPS for the Wikipedia category. The Goi-Taikei category is prefixed with ID number.

[3] The maximum depth of the common noun hierarchy is 12. Most links are *is-a* relations, but some are *part-of* relations, which are explicitly marked

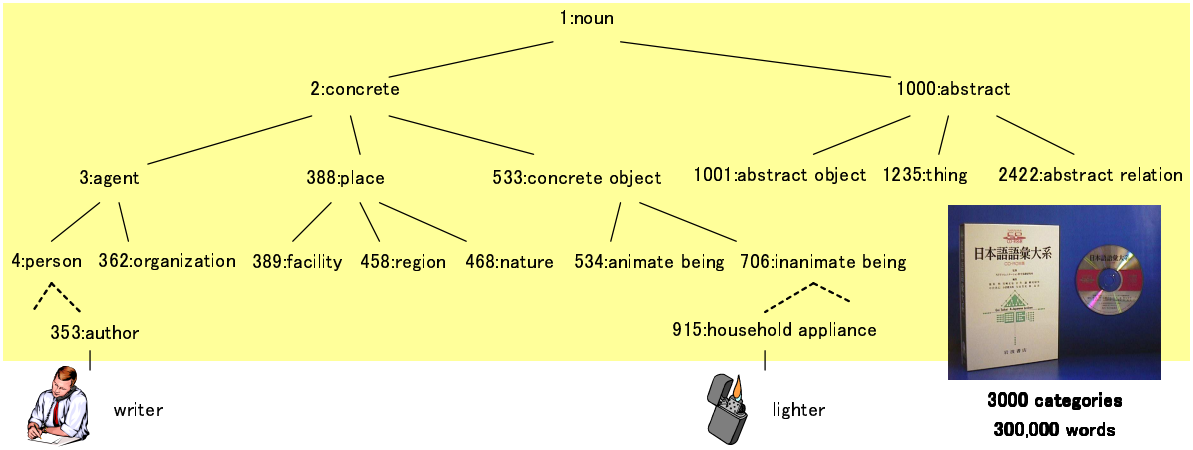

Figure 1: Top three layers of the common noun semantic category hierarchy in *Nihongo Goi-Taikei*

<title> カクテル </title> カクテル (英語:Cocktail) とは、主にベースとなる酒に、他の酒またはジュースなどを混ぜて作るアルコール飲料 ... <Category> カクテル </Category>	<title>cocktail</title> A cocktail (English:Cocktail) is an alcoholic beverage made by mixing a base liquor with other liquor or juice. ... <Category>cocktail</Category>
<title>Category:カクテル </title> [[カクテル]] に関するカテゴリ ... <Category> 酒 </Category>	<title>Category:Cocktails</title> Category on [[cocktails]] ... <Category>alcoholic beverages</Category>

Figure 2: Examples of title, body (definition sentence), and category for article page and category page in Japanese Wikipedia (left) and their translation (right)

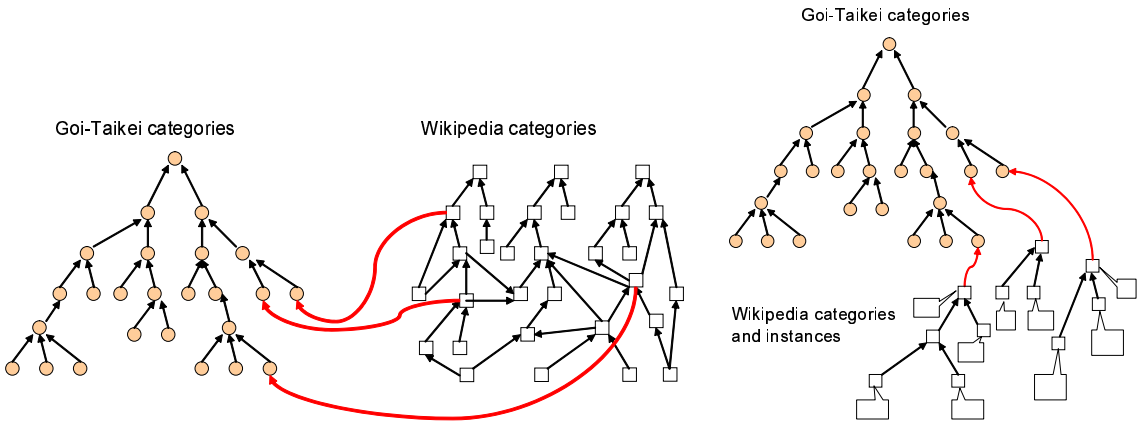

Figure 3: The ontology building method: First, Goi-Taikei leaf categories are aligned with Wikipedia categories (left), then each leaf category is extended by detecting *is-a* links in Wikipedia (right).

3.1 Category Alignment

For each leaf category in Goi-Taikei, we first make a list of junction category candidates. Wikipedia categories satisfying at least one of the following three conditions are extracted as candidates:

- The Goi-Taikei category name exactly matches the Wikipedia category name.
- One of the instances of the Goi-Taikei category exactly matches the Wikipedia category name.
- More than two instances of the Goi-Taikei category exactly match either instances or subcategories of the Wikipedia category.

Here, an instance of a Goi-Taikei category refers to words belonging to the Goi-Taikei category while that of a Wikipedia category refers to the title (name) of articles belonging to the Wikipedia category.

If a Goi-Taikei category and a Wikipedia category refer to the same concept, we regard them as semantically equivalent. If an instance of a Goi-Taikei category and a Wikipedia category refer to the same concept, we regard the name of the Goi-Taikei instance as a subcategory of the Goi-Taikei category and regard the subcategory and the Wikipedia category as semantically equivalent.

This is a sort of word sense disambiguation problem. For example, Wikipedia category ロケット (ROCKET) exactly matches the word ロケット in Goi-Taikei, which has two semantic categories, 990:aircraft (rocket) and 834:accessories (locket). Only the 990:aircraft sense of the word in Goi-Taikei matches the Wikipedia category.

We performed manual alignment because the accuracy of this category alignment is very important as regards the subsequent steps. Manual alignment is feasible and cost effective since there are only 1,921 leaves in the Goi-Taikei category hierarchy. However, we also report the result of automatic alignment in the experiment.

3.2 Hypernym Extraction

As preparation for detecting *is-a* links in the Wikipedia category network, we automatically extract a *hypernym* of the name of each article and category in advance.

We regard the first sentence of each article page as the definition of the concept referred to by the title. We applied language dependent lexico-syntactic patterns to the definition sentence to extract the hypernym. The hypernym of the category name is extracted from the definition sentence if it exists. If there is an article whose title is the same as its category, the hypernym of the article is used as that of the category.

As for lexico-syntactic patterns, we used almost the same patterns described in previous work related to Japanese such as (Kobayashi et al., 2008; Sumida et al., 2008), which is basically equivalent to work related to English such as (Hearst, 1992). Here are some examples.

[hypernym] の (一つ | 一種 | 名称 |...)
(one|kind|name|...) of [hypernym]

[hypernym](をいい | である |...)
(is_a|refers_to|...) [hypernym]

[hypernym]<EOS>
<BOS>[hypernym]

where <BOS> and <EOS> refer to the beginning and the end of a sentence.

For example, from the first article in Figure 2, the words アルコール飲料 (alcoholic beverage) are extracted as the hypernym of the article カクテル (cocktail), using the third lexico-syntactic pattern above. Since the title of the article is the same as the category name, アルコール飲料 (alcoholic beverage) is regarded as the hypernym of the category カクテル (COCKTAIL).

3.3 *Is-a* Link Detection

We automatically detect *is-a* links in the Wikipedia category network to extend the original Goi-Taikei category hierarchy. Starting from a junction category, we recursively traverse the Wikipedia category network if the link from the current category to the child category is regarded as an *is-a* link.

We regard a link between a parent category and a child category as an *is-a* link if the suffix of the child category name matches one of the *hypernym*

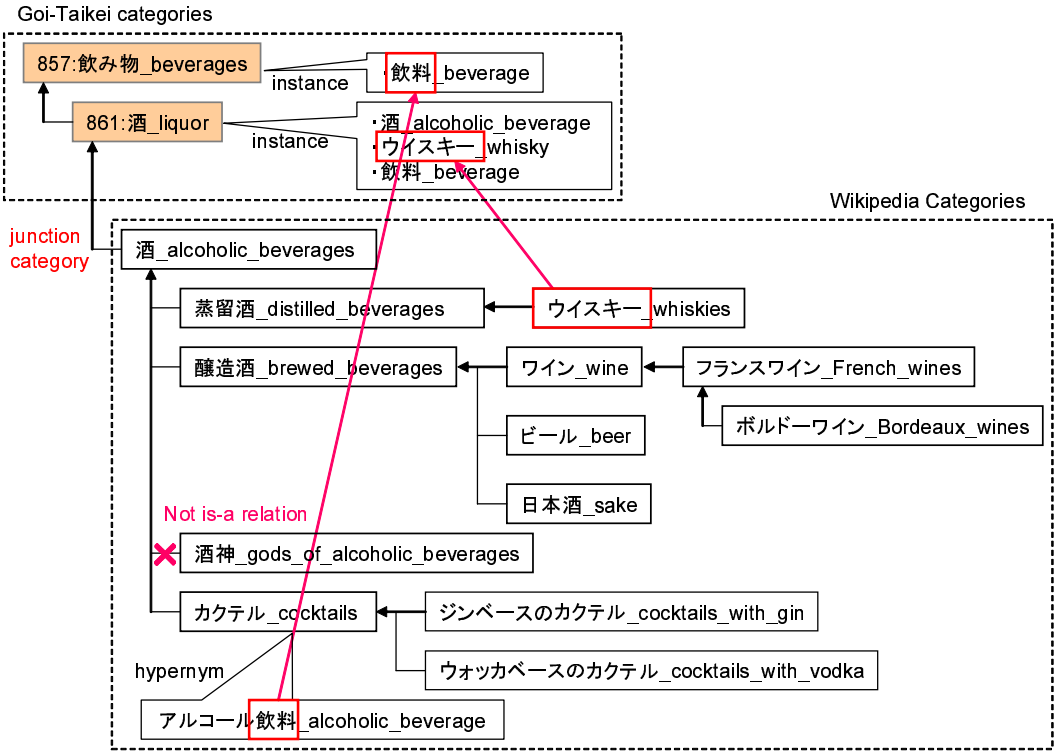

Figure 4: Extending Goi-Taikei leaf categories using the Wikipedia category network

candidates for the child category. We define the hypernym candidates for a category as the union of the following words:

- The names of three super categories in Goi-Taikei from the junction category, namely the leaf category, its parent, and its grandparent.
- All instance names belonging to the above three categories in Goi-Taikei.
- The names of all super categories in Wikipedia from the current category to the junction category.

We also regard a link as being *is-a* if the suffix of the hypernym (defined in Sec 3.2) of the child category name matches one of the hypernym candidates for the child category.

Figure 4 shows examples. The link between the category 蒸留酒 (DISTILLED BEVERAGES) and the category ウイスキー (WHISKIES) in Wikipedia is regarded as *is-a* because the word ウイスキー (whisky) is an instance of Goi-Taikei category 861:liquor just above the junction category 酒 (ALCOHOLIC BEVERAGES). The link between the category 酒 ALCOHOLIC BEVERAGES and the category カクテル (COCKTAILS) in Wikipedia is regarded as *is-a* because the suffix of アルコール飲料 (alcoholic beverage), the hypernym of the category カクテル (COCKTAILS), matches 飲料 (beverage), an instance of the category 857:beverages in Goi-Taikei. However, the link between the category 酒 (ALCOHOLIC BEVERAGES) and the category 酒神 (GODS OF ALCOHOLIC BEVERAGES) in Wikipedia is not *is-a* because the two Japanese strings do not have a common suffix.

3.4 Instance Extraction

For each Wikipedia category included in the *is-a* hierarchy constructed by the procedure described in the previous subsection, we extract the title of Wikipedia articles listed on the category page as an instance. The instance extraction method is basically the same for *is-a* category detection. We regard the link between a category and an article

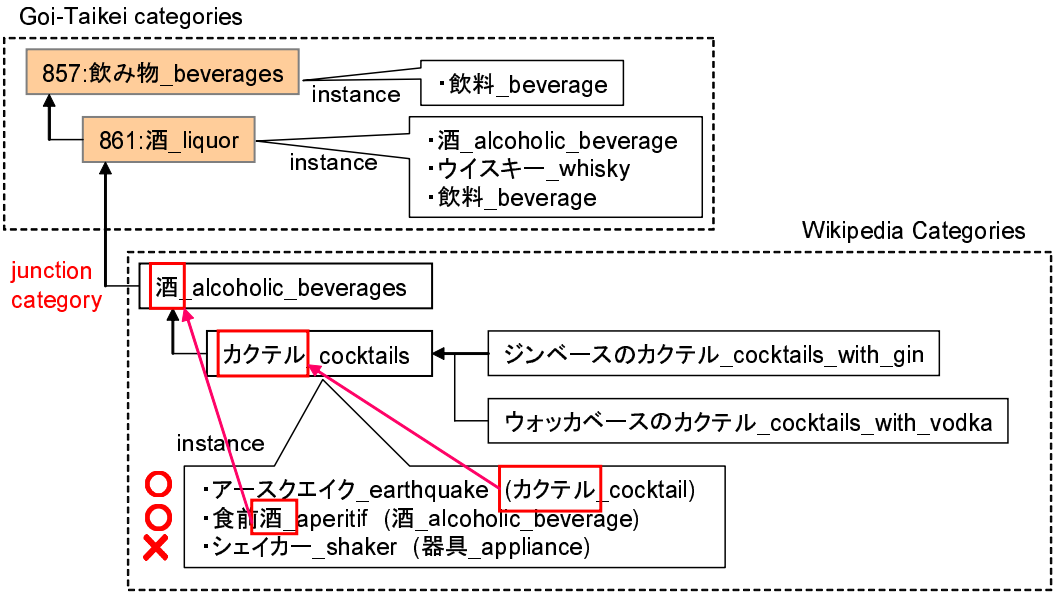

Figure 5: Extracting instances from Wikipedia category pages

as *is-a* if the suffix of either the article name or its hypernym (defined in Sec 3.2) matches one of the hypernym candidates (defined in Sec 3.3) of the article.

Figure 5 (a) shows examples. The link between the article アースクエイク (earthquake) and the category カクテル (COCKTAILS) is *is-a* because カクテル (cocktail), the hypernym of the article name アースクエイク (earthquake), exactly matches the parent category name. The link between the article 食前酒 (aperitif) and the category カクテル (COCKTAILS) is *is-a* because the suffix of 食前酒 (aperitif) matches the junction category 酒 (ALCOHOLIC BEVERAGES). The link between the article シェイカー (shaker) and the category カクテル (COCKTAILS) is not *is-a* because neither the suffix of the category name シェイカー (shaker) nor that of its hypernym 器具 (appliance) matches any hypernym candidates of the article シェイカー (shaker).

4 Experimental Result and Discussion

4.1 Category Alignment

We used the XML file of the Japanese Wikipedia as of July 24, 2008[4]. There are 49,543 category pages and 479,231 article pages in the file. For each of the 1,921 Goi-Taikei leaf categories with the total of 108,247 instances, we applied the three conditions described in Sec 3.1 and obtained 6,301 Wikipedia categories as junction category candidates. We then manually selected 2,477 categories as the junction categories. The number of Goi-Taikei leaf categories with one or more junction categories is 719 (719/1921=38.4%).

We performed some preliminary experiments on the automatic selection of junction categories. We trained an SVM classifier using the above junction category candidates and manual judgement results. Given a pair consisting of a Goi-Taikei category and a Wikipedia category, the SVM classifier predicts whether or not the two categories should be aligned. We used standard ontology mapping features (Euzenat and Shavaiko, 2007) such as whether the (class|instance) name of the (self|parent|children|siblings) match one and the other. We undertook a fivefold cross validation and obtained about 90% precision and 70% recall. The results were encouraging but we decided to use the manual alignment results for subsequent experiments.

[4] http://download.wikimedia.org/jawiki/

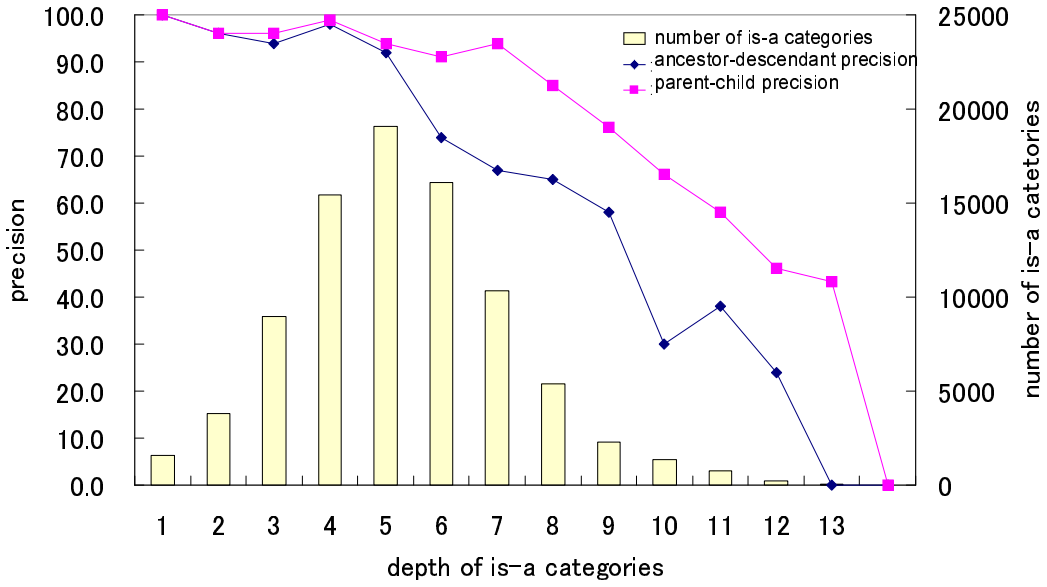

Figure 6: The precision of *is-a* links classified by the depth in the constructed category hierarchy

4.2 *Is-a* Link Detection

We extracted 23,289 categories from 49,543 categories in Wikipedia (47%) to extend the Goi-Taikei category hierarchy. We evaluated the *Is-a* link detection accuracy for the Wikipedia category network by employing the following two criteria:

- parent-child precision: whether the link between the current category and its immediate parent is an *is-a* relation.
- ancestor-descendant precision: whether all the links from the current category to the root are *is-a* relations.

We randomly selected 100 categories at each depth from the constructed hierarchy and manually evaluated the parent-child precision and the ancestor-descendant precision. Figure 6 shows the precisions of *is-a* links classified by the depth in the constructed category hierarchy. It also shows the number of categories at each depth.

The parent-child precision is more than 90% from depths 1 to 7, while the ancestor-descendant precision is more than 90% froms depth 1 to 5. After excluding depth 1 categories (junction categories whose precision is 100%), the average parent-child precision is 92.8% and the average ancestor-descendant precision is 82.6%.

4.3 Instance Extraction

We extracted 263,631 articles from 479,231 articles in Wikipedia (55%) as instances of the constructed category hierarchy. The category with the largest number of instances is 日本の俳優 (JAPANESE ACTORS) with 5,632 instances. The average number of instances for a category is 17.8.

We evaluate the accuracy of instance extraction as follows: For each category in the constructed hierarchy, we list all its articles, and construct a pair consisting of a category and an article. We randomly sample these pairs and leave only the pairs in which all the links from its category to the root are *is-a* relations by manual inspection. For 319 category-article pairs obtained by this procedure, 247 articles are manually classified as instances of the category, while 208 articles are automatically classified as instances. The intersection of the two is 205. Thus, the precision and recall of instance extraction are 98.6%(205/208) and 83.0%(205/247), respectively.

4.4 Comparison to Previous Methods

Sakurai et al. (2008) reported the parent-child precision of their suffix matching-based method was 91.2% and 6,672 Wikipedia categories are used to construct their (fragmentary) hierarchy. We used a much larger set of Wikipedia categories (23,239) to extend the Goi-Taikei to form a single unified hierarchy with a comparable parent-child precision (92.8%). Kobayashi et al. (2008) reported their alignment accuracy (parent-child precision) was 93% and 19,426 Wikipedia categories are directly aligned with Goi-Taikei categories. We used a significantly larger set of Wikipedia categories (19426/23239=0.84) to extend the Goi-Taikei with retaining the *is-a* relations included in the Wikipedia category network.

5 Conclusion

In this paper, we presented a method for building a large-scale, Japanese ontology from Wikipedia using one of the most popular Japanese thesauri, *Nihongo Goi-Taikei*, as its upper ontology. Unlike previous methods, it can create a single connected taxonomy with a well-defined upper level taxonomy inherited from Goi-Taikei, as well as a fined-grained and up-to-date lower level taxonomy with broad-coverage extracted from Wikipedia.

Future work will include automatic category alignment between Goi-Taikei and Wikipedia to fully automate the ontology building. It would be interesting to use another Japanese thesaurus, such as the recently released Japanese WordNet (Bond et al., 2008), as an upper ontology for the proposed method.

One of the problems with the proposed method is that it only uses about half of the knowledge (categories and articles) in Wikipedia. This is because we restricted the alignment points in Goi-Taikei category hierarchy to its leaves. In Ponzetto and Navigli (2009), they present a method for aligning WordNet and Wikipedia fully at many levels with both of them retaining a hierarchal structure. However, their method does not integrate the two hierarchies into a single taxonomy. We think that developing a method for merging the two hierarchies into one taxonomy is the key to extracting more information from Wikipedia.

References

Bond, Francis, Hitoshi Isahara, Kyoko Kanzaki, and Kiyotaka Uchimoto. 2008. Boot-strapping a wordnet using multiple existing wordnets. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC)*, pages 28–30.

Euzenat, Jérôme and Pavel Shavaiko. 2007. *Ontology Matching*. Springer.

Hearst, Marti A. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th Conference on Computational Linguistics (COLING)*, pages 539–545.

Ikehara, Satoru, Masahiro Miyazaki, Satoshi Shirai, Akio Yokoo, Hiromi Nakaiwa, Kentaro Ogura, Yoshifumi Ooyama, and Yoshihiko Hayashi, editors. 1997. *Nihongo Goi-Taikei – a Japanese Lexicon*. Iwanami Shoten. (in Japanese).

Kobayashi, Akio, Shigeru Masuyama, and Satoshi Sekine. 2008. A method for automatic construction of general ontology merging goi-taikei and japanese wikipedia. In *Information Processing Society of Japan (IPSJ) SIG Technical Report 2008-NL-187 (in Japanese)*, pages 7–14.

Ponzetto, Simone Paolo and Roberto Navigli. 2009. Large-scale taxonomy mapping for restructuring and integrating wikipedia. In *Proceedings of the 21st International Joint Conference of Artifi cial Intelligence (IJCAI)*, pages 2083–2088.

Ponzetto, Simone Paolo and Michael Strube. 2007. Deriving a large scale taxonomy from wikipedia. In *Proceedings of the 22nd Conference on the Advancement of Artifi cial Intelligence (AAAI)*, pages 1440–1445.

Sakurai, Shinya, Takuya Tejima, Masayuki Ishikawa, Takeshi Morita, Noriaki Izumi, and Takahira Yamaguchi. 2008. Applying japanese wikipedia for building up a general ontology. In *Japanese Society of Artifi cial Intelligence (JSAI) Technical Report SIG-SWO-A801-06 (in Japanese)*, pages 1–8.

Suchanek, Fabian M., Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge unifying wordnet and wikipedia. In *Proceedings of the 16th International Conference on World Wide Web (WWW)*, pages 697–706.

Sumida, Asuka, Naoki Yoshinaga, and Kentaro Torisawa. 2008. Boosting precision and recall of hyponymy relation acquisition from hierarchical layouts in wikipedia. In *Proceedings of the sixth Language Resources and Evaluation Conference (LREC)*, pages 28–30.

Multilingual Lexical Network from the Archives of the Digital Silk Road

Mohammad Daoud
LIG, GETALP
Université Joseph Fourier
Mohammad.Daoud@imag.fr

Christian Boitet
LIG, GETALP
Université Joseph Fourier
Christian.Boitet@imag.fr

Mathieu Mangeot
LIG, GETALP
Université Joseph Fourier
Mathieu.Mangeot@imag.fr

Kyo Kageura
Graduate School of Education
The University of Tokyo
kyo@p.u-tokyo.ac.jp

Asanobu Kitamoto
The National Institute of Informatics (Tokyo)
Kitamoto@nii.ac.jp

Abstract

We are describing the construction process of a specialized multilingual lexical resource dedicated for the archive of the Digital Silk Road DSR. The DSR project creates digital archives of cultural heritage along the historical Silk Road; more than 116 of basic references on Silk Road have been digitized and made available online. These books are written in various languages and attract people from different linguistic background, therefore, we are trying to build a multilingual repository for the terminology of the DSR to help its users, and increase the accessibility of these books. The construction of a terminological database using a classical approach is difficult and expensive. Instead, we are introducing specialized lexical resources that can be constructed by the community and its resources; we call it Multilingual Preterminological Graphs MPGs. We build such graphs by analyzing the access log files of the website of the Digital Silk Road. We aim at making this graph as a seed repository so multilingual volunteers can contribute. We have used the access log files of the DSR since its beginning in 2003, and obtained an initial graph of around 116,000 terms. As an application, We have used this graph to obtain a preterminological multilingual database that has a number of applications.

1 Introduction

This paper describes the design and development of a specialized multilingual lexical resource for the archive constructed and maintained by the Digital Silk Road project. The Digital Silk Road project (NII 2003) is an initiative started by the National Institute of Informatics (Tokyo/Japan) in 2002, to archive cultural historical resources along the Silk Road, by digitizing them and making them available and accessible online.

One of the most important sub-projects is the Digital Archive of Toyo Bunko Rare Books (NII 2008) where 116 (30,091 pages) of old rare books available at Toyo Bunko library have been digitized using OCR (Optical Character Recognition) technology. The digitized collection contains books from nine languages including English. The website of the project attracts visitors from the domain of history, archeology, and people who are interested in cultural heritage. It provides services of reading and searching the books of Toyo Bunko, along with variety of services. Table 1 shows the countries from which DSR is being accessed. The table

Proceedings of the 6th Workshop on Ontologies and Lexical Resources (Ontolex 2010), pages 19–27,
Beijing, August 2010

shows that around 60% of visitors are coming from countries other than Japan. The diversity of the visitors' linguistic backgrounds suggests two things: 1) Monolingual translation service is not enough. 2) It shows that we can benefit from allowing them to contribute to a multilingual repository. So we design and build a collaborative multilingual terminological database and seed using the DSR project and its resources (Daoud, Kitamoto et al. 2008). However, Discovering and translating domain specific terminology is a very complicated and expensive task, because (1) traditionally, it depends on human terminologists (Cabre and Sager 1999) which increases the cost, (2) terminology is dynamic (Kageura 2002), thousands of terms are coined each year, and (3) it is difficult to involve domain experts in the construction process. That will not only increase the cost, but it will reduce the quality, and the coverage (number of languages and size). Databases like (UNGeo 2002; IATE 2008; UN 2008) are built by huge organizations, and it is difficult for a smaller community to produce its own multilingual terminological database.

Country	Visitors	language	Books in the same language
Japan	117782	JA	2 books
China	30379	CH	5 books
USA	15626	EN	44 books
Germany	8595	GE	14 books
Spain	7076	SP	-
Australia	5239	EN	See USA
Italy	4136	IT	1 book
France	3875	FR	14 books
Poland	2236	PO	-
Russia	1895	RU	7 books
other	87573	Other	There are many books in different language
Total	284412		

Table 1. Countries of the DSR visitors (from jan/2007 to dec/2008)

In the next section we will give definitions for the basic concepts presented in this article, in particular, the preterminology and its lexical network (graph). Then, in the third section we will show the automatic approach to seed the multilingual preterminological graph based on the resources of the DSR. And then, we will discuss the human involvement in the development of such a resource by providing a study of the possible contributors through analyzing the multilinguality and loyalty of the DSR visitors. In the fifth section we will show the experimental results. And finally, we will draw some conclusions.

2 Multilingual Preterminological Graphs

2.1 Preterminology

Terminological sphere of a domain is the set of terms related to that domain. A smaller set of that sphere is well documented and available in dictionaries and terminological databases such as (FAO 2008; IEC 2008; IDRC 2009)... However, the majority of terms are not multilingualized, nor stored into a database, even though, they may be used and translated by the community and domain experts. This situation is shown in Figure 1, where the majority of terms are in area **B**. Preterminological sphere (area **B**) of a domain is a set of terms (*preterms*) related to the domain and used by the community but it might not be documented and included in traditional lexical databases.

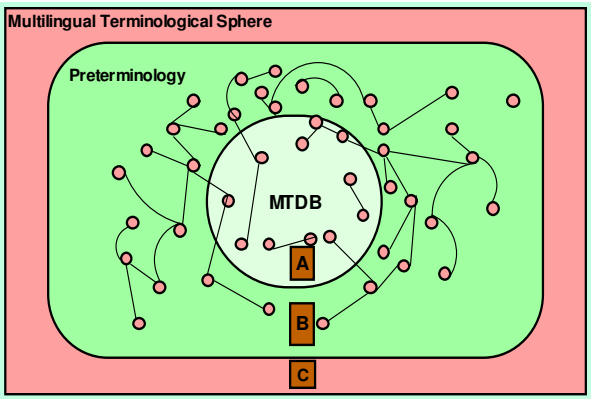

Figure 1. Preterminological sphere

Every year thousands of terms are coined and introduced in correspondence to new concepts, scientific discoveries or social needs. Most of these terms are produced in the top dominant languages, i.e. English. Interested people from different linguistic backgrounds would find suitable translations to new terms and use it amongst them. For example, the term 'status update' is used by people who visit social networking websites like facebook.com. Translation of this term to Arabic might not be available in area **A** of Figure 1. However the Arabic community found a translation that is acceptable which is تحديث الحالة. So this term is in the area **B**. We are trying to use what is in area **A**, and what can be contributed from **B** to build preterminology (Daoud, Boitet et al. 2009).

2.2 Structure of MPG

We are building preterminological resource as a lexical network (graph) to handle the diversity of the resources that we use. A multilingual preterminological graph *MPG(N,E)* is a finite non-empty set *N={n1,n2, …}* of objects called Nodes together with a set *E={e1,e2, …}* of unordered pairs of distinct nodes of *MPG* called edges. This definition is based on the general definition of a graph at the following references (Even 1979; Loerch 2000). *MPG* of domain *X*, contains possible multilingual terms related to that domain connected to each other with relations. A multilingual lexical unit and its translations in different languages are represented as connected nodes with labels.

In an *MPG* the set of nodes **N** consists of *p,l, s, occ,* where *p* is the string of the preterm, *l* is the language, *s* is the code of the first source of the preterm, and *occ* is the number of occurrences. Note that *l* could be undefined. For example: *N={[silk road, en, log],[Great Wall of China, en, ,wikipedia, 5], [الصين, ar, contributorx,6]}*, here we have three nodes, 2 of them are English and one in Arabic, each term came from a different source. Note that English and Arabic terms belong to the same *N* thus, the same MPG.

An *Edge e={n, v}* is a pair of nodes adjacent in an *MPG*. An edge represents a relation between two preterms represented by their nodes. The nature of the relation varies. However, edges are weighted with several weights (described below) to indicate the possible nature of this relation.

The following are the weights that label the edges on an MPG: *Relation Weights rw*: For an edge *e={[p1,l1,s1], [p2,l2,s2]}, rw* indicates that there is a relation between the preterm *p1* and *p2*. The nature of the relation could not be assumed by *rw*. *Translation Weights tw*: For an edge *e={[p1,l1,s1], [p2,l2,s2]}, tw* suggests that *p1* in language *l1* is a translation of *p2* in language *l2*. *Synonym Weights sw*: For an edge *e={[p1,l1,s1], [p2,l1,s2]}, sw* suggests that *p1* and *p2* are synonyms.

3 Automatic Initialization of DSR-MPG

Basically we seeded DSR-MPG, through two steps, the firs one is the automatic seeding, which consists of the following: 1) Initialization by finding interesting terms used to search the website of the DSR. 2) Multilingualization, using online resources. 3) Graph Expansion using the structure of the graph it self. The second step is the progressive enhancement, by receiving contributions from users, through set of useful applications. In this section we will discuss the first three steps. In section 4, we will discuss the human factor in the development of DSR-MPG.

3.1 Analyzing Access Log Files

We analyze two kinds of access requests that can provide us with information to enrich the MPG: (1) requests made to the local search engine of DSR (2) requests from web-based search engine (like Google, Yahoo!…). These requests provide the search terms that visitors used to access the website. Moreover, we can understand the way users interpret a concept into lexical units. For example, if we find that five different users send two search requests *t1* and *t2*, then there is a possibility that t1 and *t2* have a relation. The graph constructor analyzes the requests to make the initial graph by creating edges between terms in the same session. *rw(x,y)*, is set to the number of sessions containing *x* and *y* within the log file.

For example, *rw(x,y)* = 10 means that 10 people thought about *x* and y within the same search session. Figure 2 shows an example of a produced graph. The method did not discover the kind of relation between the terms. But it discovered that there is a relation, for example, three users requested results for "*yang*" followed by "*yin*" within the same session. Hence, edge with weight of 2 was constructed based on this.

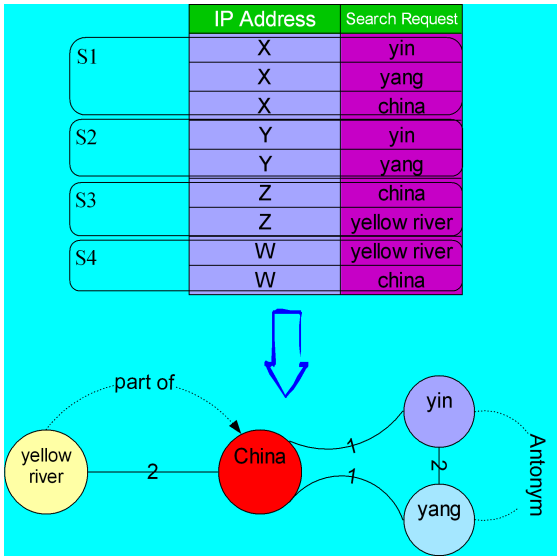

Figure 2. Example of constructing an MPG from an access log file

3.2 Multilingualization Using Online Resources

Many researchers focused on the usage of dictionaries in digital format to translate lexical resources automatically (Gopestake, Briscoe et al. 1994) (Etzioni, Reiter et al. 2007). We are concerned with the automatic utilization of these resources to acquire multilingual preterminological resources through the following: 1) Wikipedia 2) online MT systems 3) online dictionaries.

Wikipedia (Wikipedia-A 2008) is a rich source of preterminology, it has good linguistic and lexical coverage. As of December, 2009, there are 279 Wikipedias in different languages, and 14,675,872 articles. There are 29 Wikipedias with more that 100000 articles and 91 languages have more than 10,000 articles. Beside, Wikipedia is built by domain experts. We exploit the structure of Wikipedia to seed an MPG, by selecting a root set of terms, for each one of them we fetch its wikipedia article, and then we use the language roll of the article. For example, we fetch the article (Cuneiform script) En: http://en.wikipedia.org/wiki/Cuneiform_script, to reach its translation in Arabic from this url: http://ar.wikipedia.org/wiki/كتابة_مسمارية

We use also online machine translation systems as general purpose MRDs. One of the main advantages of MT systems is the good coverage even for multiword terms. The agreement of some MT systems with other resources on the translation of one term enhanced the confidence of the translation. Another positive point is that the results of MT provide a first draft to be post edited later. We used 3 MT systems:

- Google Translate (Google 2008) (50 languages)
- Systran (Systran 2009) (14 languages)
- Babylon (Babylon 2009) (26 languages)

Here is an example of translating the term "great wall of China" into Arabic.

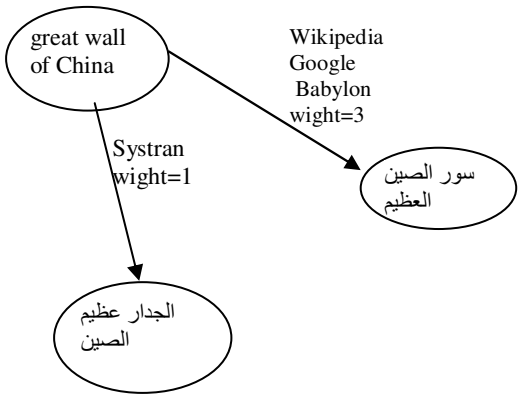

Figure 3. MPG sample nodes

In a similar way, we used several online repositories; to make good use of what is available and standardized, to initializing the MPG with various resources, and to construct a meta-system to call online dictionaries automatically. We used IATE (IATE 2008) as an example of a terminological db, and Google dictionary (Google 2008). The concept is similar to the concept of using online translations, where we construct an http request, to receive the result as html page.

3.3 Graph Expansion

And then, the Graph is expanded by finding the synonyms according to formula (1) described at (Daoud, Boitet et al. 2009). After finding synonyms we assume that synonyms share the same translations. As Figure 4 shows, *X1* and *X2* have translations overlaps, and relatively high *rw*, so that suggest they are synonyms. Therefore we constructed heuristic edges between the translations of *X1* and *X2*.

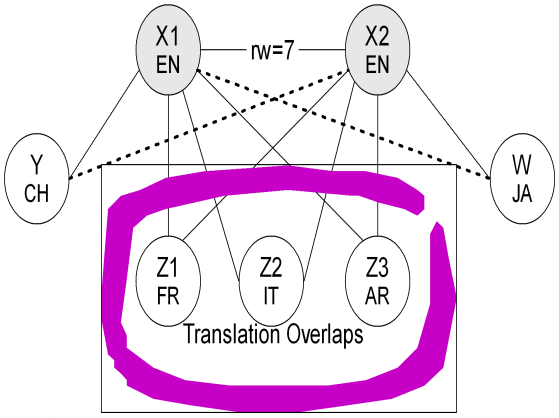

Figure 4. Graph expansion

4 Human Involvement in the Development of DSR-MPG

After initializing the graph, we target contributions from the visitors to the DSR website. In this section we will start by analyzing the possibility of receiving contributions from the visitors, and then we will introduce some useful applications on the DSR-MPG that can help the visitors and attract them to get involved.

4.1 Analyzing Possible Contributors of the DSR

We are trying to analyze access log files to find out the possible contributors to a preterminological multilingual graph dedicated to an online community. This kind of information is necessary for the following reasons: 1) it provide feasibility analysis predicting the possibility of receiving contribution to a multilingual preterminological repository. 2) it gives information that can be used by the collaborative environment to personalize the contribution process for those who prove to be able to contribute.

In the analysis process we are using the following information that can be easily extracted the access records:

- Key terms to access the historical resources of the Digital Silk Road, whether it is the local search engine, or any external search engine.
- Access frequency: number of access requests by a visitor over a period of time.
- Language preferences
- Period of visits

Knowing these points helps determining the possible users who might be willing to contribute. A contributor should satisfy the following characteristics: 1) *Loyalty* 2) *Multilinguality*. A multilingual user is a visitor who uses multilingual search terms to access the online resources. We rank users based on their linguistic competence, we measure that by tracking users' search requests, and matching them with the multilingual preterminological graph, users with higher matches in certain pair of languages are ranked higher. A *loyal user* is a user who visits the web site frequently and stays longer than other users. Users based on how many months they accessed the website more that k times.

4.2 DSR-MPG Applications

For a historical archive like the DSR, we find that reading and searching where the most important for users. Log files since 2003 shows that 80% of the project visitors were interested in reading the historical records. Moreover, around 140000 search requests have been sent to the internal search engine. So we implemented two applications (1) "*contribute-while-reading*" and (2) "*contribute-while-searching*".

4.2.1 Contribute While Searching

Physical books have been digitized and indexed into a search engine. We expect users to send monolingual search requests in any language supported by our system to get multilingual answers. Having a term base of multilingual equivalences could achieve this (Chen 2002). A bilingual user who could send a bilingual search request could be a valid candidate to contribute. We plan that users who use our search engine will use the DSR-pTMDB to translate their requests and will contribute to the graph spontaneously. As Figure 5 shows, a user would translate the search request, during the searching process; the user can ask to add new translation if s/he was not happy with the suggested translation, by clicking on "Add Suggestions" to view a contribution page.

Figure 5. A Japanese user translating his request

4.2.2 Contribute While Reading

The other application is trying to help users from different linguistic backgrounds to translate some of the difficult terms into their languages while they are reading, simply by selecting a term from the screen. As shown in Figure 6, readers will see a page from a book as an image, with its OCR text. Important terms will be presented with yellow background. Once a term is clicked, a small child contribution/lookup window will be open, similar. Also user can lookup/translate any term from the screen by selecting it. This application helps covering all the important terms of each book.

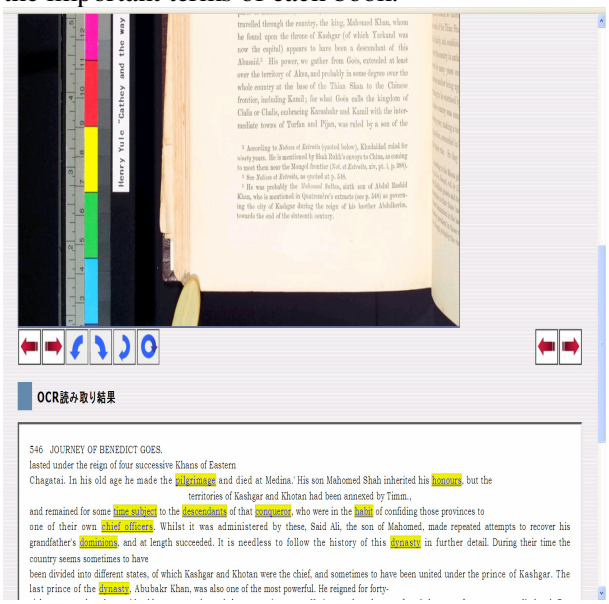

Figure 6. Translate while reading

5 Experimental Results

In this section present we will present the experiment of seeding DSR-MPG, and the results of discovering possible contributors from the visitors of the DSR.

5.1 DSR-MPG Initialization

To build the initial DSR-MPG, we used the access log files of the DSR website (dsr.nii.ac.jp) from December 2003 to January 2009. The initial graph after normalization contained 89,076 nodes. Also we extracted 81,204 terms using Yahoo terms. 27,500 of them were not discovered from the access files. So, the total number of nodes in the initial graph was 116,576 nodes, see Figure 7 for sample nodes.

After multilingualization, the graph has 210,781 nodes containing terms from the most important languages. The graph has now 779,765 edges with $tw > 0$. The important languages are the languages of the majority of the visitors, the languages of the archived books, and representative languages a long the Silk Road. DSR-MPG achieved high linguistic coverage as 20 languages have more than 1000 nodes on the graph. To evaluate the produced graph, we extracted 350 English terms manually from the index pages of the following books:

Ancient Khotan, vol.1: http://dsr.nii.ac.jp/toyobunko/VIII-5-B2-7/V-1/

On Ancient Central-Asian Tracks, vol.1:http://dsr.nii.ac.jp/toyobunko/VIII-5-B2-19/V-1

Memoir on Maps of Chinese Turkistan and Kansu, vol.1: http://dsr.nii.ac.jp/toyobunko/VIII-5-B2-11/V-1

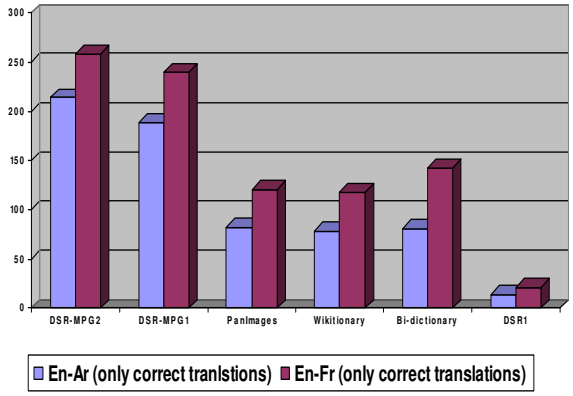

Figure 7. A comparison between DSR-MPG, and other dictionaries. The En-Ar bi-dictionary is Babylon (Babylon 2009), and the En-Fr bi-dictionary was IATE.

We assume that the terms available in these books are strongly related to the DSR. Hence, we tried to translate them into Arabic and French. Figure 7 compares between DSR-MPG, and various general purpose dictionaries. Out of the 350 terms, we found 189 correct direct translations into Arabic. However, the number reached 214 using indirect translations. On the other hand, the closest to our result was PanImages, which uses Wikitionaries and various dictionaries, with only 83 correct translations.

DSR-MPG1 is the translations obtained from formula 1, DSR-MPG2 represents the translations obtained from indirect translations, which increased the amount of correct translation by

25 terms in the case of En-Ar. The result can be progressively enhanced by accepting contributions from volunteers through the applications we described in the section three and the generic nature of MPG makes it easy to accept contributions from any dictionary or terminological database.

Around 55200 root English terms were used as a seed set of terms; these terms were selected from the initial DSR-MPG. Around 35000 terms have been translated from Wikipedia into at least 1 language, mostly in French, German. Wikipedia increased the density of the graph by introducing around 113,000 edges (with *tw*).

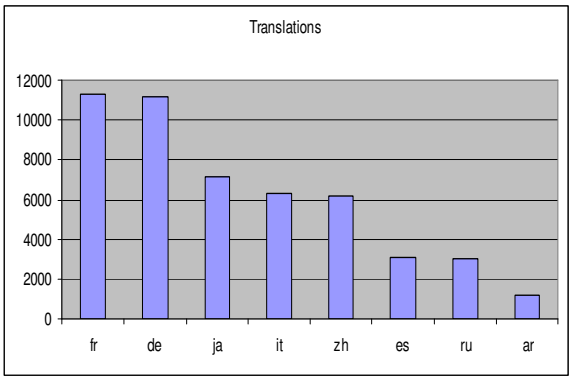

Figure 8. Number of translated terms in sample languages using Wikipedia

Naturally MT would achieve better coverage; we checked the results for Arabic, we selected 60 terms randomly from the root set, around 25 terms were translated correctly. 13 terms needed slight modification to be correct.

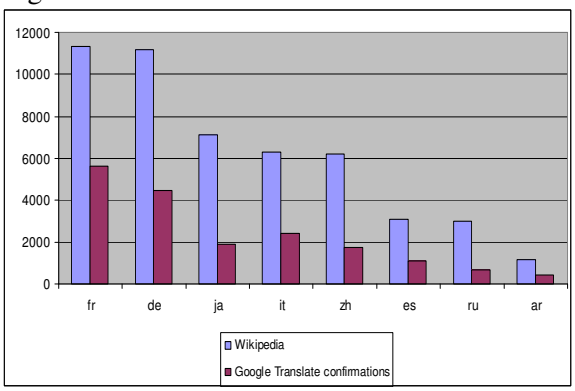

Figure 9. Terms translated by Google MT and matched the translation of Wikipedia

5.2 DSR Possible Contributors

With K=2, meaning that a *multilinguality competence* is counted only if the two terms sent by a user has to have more than 2 points of translation weight on the MPG.

The highest score was 33, achieved by this IP: p27250-adsao05douji-acca.osaka.ocn.ne.jp. That means that this user sent 33 multilingual search requests. We have another 115 users with score higher than 5.

For example, the following two request, sent by one user:

p27250-adsao05douji-acca.osaka.ocn.ne.jp
&input=peshawar
p27250-adsao05douji-acca.osaka.ocn.ne.jp
&input=ペシャワール

On the DSR-MPG the translation weight between peshawer and ペシャワール = 5, thus this IP earned a point. With k=10, means that a user should send 10 requests to earn a *loyalty point*, only 309 users earned 12 point (for 12 months), 43 of them has more than 3 points.

6 Conclusions

We presented our work in constructing a new lexical resource that can handle multilingual terms based on the historical archive of the Digital Silk Road. Multilingual Preterminological Graphs (MPGs) are constructed based on domain dedicated resources, and based on volunteer contributions.

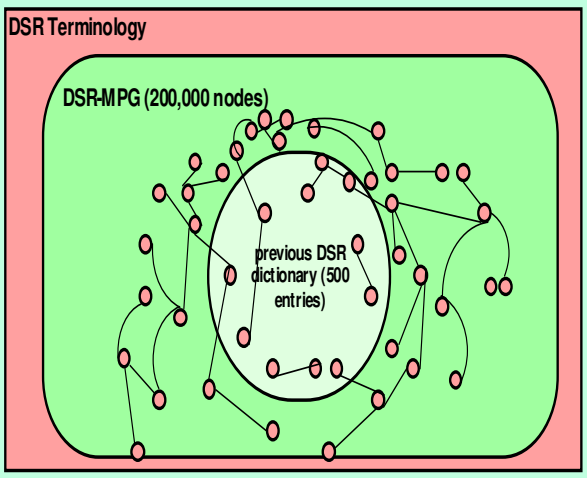

Figure 10. DSR preterminology

It compiles terms available in the preterminological sphere of a domain. In this article we defined the framework of the construction of preterminology, and we described the approach for using access log files to initialize such preterminological resource by finding the trends in the search requests used to access the resources of an online community. Aiming at a standardized multilingual repository is very expensive

and difficult. Instead of that, MPGs tries to use all available contributions. This way will enhance the linguistic and informational coverage, and tuning the weights (*tw*, *rw*, and *sw*) will give indications for the confidence of the translation equivalences, as the *tedges* accumulate the agreements of the contributors and MDRs (online resources).

We used the resources of the Digital Silk Road Project to construct a DSR-MPG and some applications that attract further contribution to the MPG. DSR-MPG achieved high linguistic and informational coverage compared to other general purpose dictionaries, Figure 10. Furthermore, the generic structure of the MPG makes it possible to accept volunteer contributions, and it facilitates further study of computing more lexical functions and ontological relations between the terms. We made a study on the possibility of receiving contributions from users, by analyzing the access log file to find multilinguality and loyalty of the DSR visitors; we found 115 users with the needed linguistic capacity 43 of them scored high loyalty points. This gives an indication of the future of the contributions. These measures are just estimations and expected to go high with the help of the MPG-DSR applications.

References

Babylon. (2009). "Babylon Dictionary." Retrieved 5/5/2009, 2009, from http://www.babylon.com/define/98/English-Arabic-Dictionary.html.

Cabre, M. T. and J. C. Sager (1999). Terminology: Theory, methods, and applications, J. Benjamins Pub. Co.

Chen, A. (2002). "Cross-Language Retrieval Experiments at CLEF 2002." in CLEF-2002 working notes,.

Daoud, M., C. Boitet, et al. (2009). Constructing multilingual preterminological graphs using various online-community resources. the Eighth International Symposium on Natural Language Processing (SNLP2009), Thailand.

Daoud, M., C. Boitet, et al. (2009). Building a Community-Dedicated Preterminological Multilingual Graphs from Implicit and Explicit User Interactions. Second International Workshop on REsource Discovery (RED 2009), co-located with VLDB 2009, Lyon, France.

Daoud, M., A. Kitamoto, et al. (2008). A CLIR-Based Collaborative Construction of Multilingual Terminological Dictionary for Cultural Resources. Translating and the Computer 30, London-UK.

Etzioni, O., K. Reiter, et al. (2007). Lexical translation with application to image searching on the web. MT Summit XI, Copenhagen, Denmark.

Even, S. (1979). Graph Algorithms, Computer Science Press.

FAO. (2008). "FAO TERMINOLOGY." Retrieved 1/9/2008, 2008, from http://www.fao.org/faoterm.

Google. (2008). "Google Dictionary." Retrieved 1/9/2008, 2008, from http://www.google.com/dictionary.

Google. (2008). "Google Translate." Retrieved 1 June 2008, 2008, from http://translate.google.com.

Gopestake, A., T. Briscoe, et al. (1994). "Acquisition of lexical translation relations from MRDS." Machine Translation Volume 9, Numbers 3-4 / September, 1994: 183-219.

IATE. (2008). "Inter-Active Terminology for Europe." Retrieved 10/10/2008, 2008, from http://iate.europa.eu.

IDRC. (2009, 10 January 2009). "The Water Demand Management Glossary (Second Edition)." from http://www.idrc.ca/WaterDemand/IDRC_Glossary_Second_Edition/index.html.

IEC. (2008). "Electropedia." Retrieved 10/10/2008, 2008, from http://dom2.iec.ch/iev/iev.nsf/welcome?openform.

Kageura, K. (2002). The Dynamics of Terminology: A descriptive theory of term formation and terminological growth.

Loerch, U. (2000). An Introduction to Graph Algorithms Auckland, New Zealand, University of Auckland.

NII. (2003). "Digital Silk Road." Retrieved 1/9/2008, 2008, from http://dsr.nii.ac.jp/index.html.en.

NII. (2008). "Digital Archive of Toyo Bunko Rare Books." Retrieved 1 June 2008, 2008, from http://dsr.nii.ac.jp/toyobunko/.

Systran. (2009). "Systran Web Tranlstor." Retrieved 20/12/2009, 2009, from www.systransoft.com/.

UN-Geo (2002). Glossary of Terms for the Standardization of Geographical Names, UN, New York.

UN. (2008). "United Nations Multilingual Terminology Database." Retrieved 10/10/2008, 2008, from http://unterm.un.org/.

Wikipedia-A. (2008). "Wikipedia." Retrieved 1 June 2008, 2008, from http://www.wikipedia.org/.

Finding Medical Term Variations using Parallel Corpora and Distributional Similarity

Lonneke van der Plas
Department of Linguistics
University of Geneva
lonneke.vanderplas@unige.ch

Jörg Tiedemann
Department of Linguistics and Philology
Uppsala University
jorg.tiedemann@lingfil.uu.se

Abstract

We describe a method for the identification of medical term variations using parallel corpora and measures of distributional similarity. Our approach is based on automatic word alignment and standard phrase extraction techniques commonly used in statistical machine translation. Combined with pattern-based filters we obtain encouraging results compared to related approaches using similar data-driven techniques.

1 Introduction

Ontologies provide a way to formally represent knowledge, for example for a specific domain. Ontology building has received a lot of attention in the medical domain. This interest is reflected in the existence of numerous medical ontologies, such as the Unified Medical Language System (UMLS) (McCray and Hole, 1990) with its metathesaurus, semantic network, and specialist lexicon. Although the UMLS includes information for languages other than English, the coverage for other languages is generally smaller.

In this paper we describe an approach to acquire lexical information for the Dutch medical domain automatically. In the medical domain variations in terminology often include multi-word terms such as *aangeboren afwijking* 'birth defect' for *congenitale aandoening* 'congenital disorder'. These multiple ways to refer to the same concept using distinct (multi-word) terms are examples of synonymy[1] but are often referred to as term variations. These term variations could be used to enhance existing medical ontologies for the Dutch language.

Our technique builds on the distributional hypothesis, the idea that semantically related words are distributed similarly over contexts (Harris, 1968). This is in line with the Firthian saying that, 'You shall know a word by the company it keeps.' (Firth, 1957). In other words, you can grasp the meaning of a word by looking at its contexts.

Context can be defined in many ways. Previous work has been mainly concerned with the syntactic contexts a word is found in (Lin, 1998; Curran, 2003). For example, the verbs that are in a subject relation with a particular noun form a part of its context. In accordance with the Firthian tradition these contexts can be used to determine the semantic relatedness of words. For instance, words that occur in a object relation with the verb *to drink* have something in common: they are liquid. Other work has been concerned with the bag-of-word context, where the context of a word are the words that are found in its proximity (Wilks et al., 1993; Schütze, 1992).

Yet another context, that is much less studied, is the translational context. The translational context of a word is the set of translations it gets in other languages. For example, the translational context of *cat* is *kat* in Dutch and *chat* in French. This requires a rather broad understanding of the term context. The idea is that words that share a large number of translations are similar. For example both *autumn* and *fall* get the translation *herfst* in Dutch, *Herbst* in German, and *automne* in French. This indicates that *autumn* and *fall* are synonyms.

[1] Spelling variants are a type of term variations that are not included in the definition of synonymy.

Proceedings of the 6th Workshop on Ontologies and Lexical Resources (Ontolex 2010), pages 28–37,
Beijing, August 2010

A straightforward place to start looking for translational context is in bilingual dictionaries. However, these are not always publicly available for all languages. More importantly, dictionaries are static and therefore often incomplete resources. We have chosen to automatically acquire word translations in multiple languages from text. Text in this case should be understood as multilingual parallel text. Automatic alignment gives us the translations of a word in multiple languages. The so-called *alignment-based distributional methods* described in Van der Plas (2008) apply the translational context for the discovery of single word synonyms for the general domain. Any multilingual parallel corpus can be used for this purpose. It is thus possible to focus on a special domain, such as the medical domain we are considering in this paper. The automatic alignment provides us also with domain-specific frequency information for every translation pair, which is helpful in case words are ambiguous.

Aligned parallel corpora have often been used in the field of word sense discovery, the task of discriminating the different senses words have. The idea behind it is that a word that receives different translations might be polysemous. For example, a word such as *wood* receives the translation *woud* and *hout* in Dutch, the former referring to an area with many trees and the latter referring to the solid material derived from trees. Whereas this type of work is all built upon the divergence of translational context, i.e. one word in the source language is translated by many different words in the target language, we are interested in the convergence of translations, i.e. two words in the source language receiving the same translation in the target language. Of course these two phenomena are not independent. The alleged conversion of the target language might well be a hidden diversion of the source language. Since the English word might be polysemous, the fact that *woud* and *hout* in Dutch are both translated in English by *wood* does not mean that *woud* and *hout* in Dutch are synonyms. However, the use of multiple languages overshadows the noise resulting from polysemy (van der Plas, 2008).

Van der Plas (2008) shows that the way the context is defined influences the type of lexico-semantic knowledge that is discovered. After gold standard evaluations and manual inspection the author concludes that when using translational contexts more tight semantic relations such as synonymy are found whereas the conventional syntax-based approaches retrieve hypernyms, co-hyponyms, and antonyms of the target word. The performance on synonym acquisition when using translational contexts is almost twice as good as when using syntactic contexts, while the amount of data used is much smaller. Van der Plas (2008) ascribed the fact that the syntax-based method behaves in this way to the fact that loosely related words, such as *wine* and *beer*, are often found in the same syntactic contexts. The alignment-based method suffers less from this indiscriminant acceptance because words are typically translated by words with the same meaning. The word *wine* is typically not translated with a word for *beverage* nor with a word for *beer*, and neither is *good* translated with the equivalence of *bad*.

In this paper we are concerned with medical term variations that are in fact (multi-word) synonyms. We will use the translational context to compute similarity between terms. The translational context is not only very suitable to find tight relations between words, the transition from single-word synonyms to multi-word term variations is also straightforward due to advances in phrase-based machine translation. We will use word alignment techniques in combination with phrase extraction techniques from statistical machine translation to extract phrases and their translations from a medical parallel corpus. We combine this approach with Part-of-Speech (PoS) patterns from the term extraction literature to extract candidate terms from the phrase tables. Using similarity measures used in distributional methods we finally compute ranked lists of term variations.

We already noted that these term variations could be used to enhance existing ontologies for the Dutch language. On top of that we believe that the multi-lingual method that uses translations of multi-word terms in several languages could be used to expand resources built for English with translations in other languages (semi-) automatically. This last point falls outside the scope of this paper.

In the following section we will describe the alignment-based approaches to distributional similarity. In section 3 we will describe the methodology we followed in this paper in detail. We describe our evaluation in section 4 and discuss the results in section 5. Section 6 concludes this paper.

2 Alignment-based methods

In this section we explain the alignment-based approaches to distributional similarity. We will give some examples of translational context and we will explain how measures serve to determine the similarity of these contexts. We end this section with a discussion of related work.

2.1 Translational context

The translational context of a word or a multi-word term is the set of translations it gets in other languages. For the acquisition of translations for the Dutch medical terms we rely on automatic word alignment in parallel corpora.

Figure 1: Example of bidirectional word alignments of two parallel sentences

Figure 1 illustrates the automatic word alignment between a Dutch and an English phrase as a result of using the IBM alignment models (Brown et al., 1993) implemented in the open-source tool GIZA++ (Och, 2003). The alignment of two texts is bi-directional. The Dutch text is aligned to the English text and vice versa (dotted lines versus continuous lines). The alignment models produced are asymmetric. Several heuristics exist to combine directional word alignments which is usually called "symmetrization". In order to cover multi-word terms standard phrase extraction techniques can be used to move from word alignment to linked phrases (see section 3.2 for more details).

2.2 Measures for computing similarity

Translational co-occurrence vectors are used to find distributionally similar words. For ease of reading, we give an example of a single-word term *kat* in Table 1. In our current setting the terms can be both single- or multi-word terms such as *werkzame stof* 'active ingredient'. Every cell in the vector refers to a particular translational co-occurrence type. For example, *kat* 'cat' gets the translation *Katze* in German. The value of these cells indicate the number of times the co-occurrence type under consideration is found in the corpus.

Each co-occurrence type has a cell frequency. Likewise each head term has a row frequency. The row frequency of a certain head term is the sum of all its cell frequencies. In our example the row frequency for the term *kat* 'cat' is 65. Cut-offs for cell and row frequency can be applied to discard certain infrequent co-occurrence types or head terms respectively.

	DE Katze	FR chat	IT gatto	EN cat	total
kat	17	26	8	13	64

Table 1: Translational co-occurrence vector for *kat* ('cat') based on four languages

The more similar the vectors are, the more distributionally similar the head terms are. We need a way to compare the vectors for any two head terms to be able to express the similarity between them by means of a score. Various methods can be used to compute the distributional similarity between terms. We will explain in section 3 what measures we have chosen in the current experiments.

2.3 Related work

Multilingual parallel corpora have mostly been used for tasks related to word sense disambiguation such as separation of senses (Resnik and Yarowsky, 1997; Dyvik, 1998; Ide et al., 2002).

However, taking sense separation as a basis, Dyvik (2002) derives relations such as synonymy and hyponymy by applying the method of semantic mirrors. The paper illustrates how the method works. First, different senses are identified on the basis of manual word translations in sentence-aligned Norwegian-English data (2,6 million words in total). Second, senses are grouped in semantic fields. Third, features are

assigned on the basis of inheritance. Lastly, semantic relations such synonymy and hyponymy are detected based on intersection and inclusion among feature sets .

Improving the syntax-based approach for synonym identification using bilingual dictionaries has been discussed in Lin et al. (2003) and Wu and Zhou (2003). In the latter parallel corpora are also applied as a reference to assign translation likelihoods to candidates derived from the dictionary. Both of them are limited to single-word terms.

Some researchers employ multilingual corpora for the automatic acquisition of paraphrases (Shimota and Sumita, 2002; Bannard and Callison-Burch, 2005; Callison-Burch, 2008). The last two are based on automatic word alignment as is our approach.

Bannard and Callison-Burch (2005) use a method that is also rooted in phrase-based statistical machine translation. Translation probabilities provide a ranking of candidate paraphrases. These are refined by taking contextual information into account in the form of a language model. The Europarl corpus (Koehn, 2005) is used. It has about 30 million words per language. 46 English phrases are selected as a test set for manual evaluation by two judges. When using automatic alignment, the precision reached without using contextual refinement is 48.9%. A precision of 55.3% is reached when using context information. Manual alignment improves the performance by 26%. A precision score of 55% is attained when using multilingual data.

In a more recent publication Callison-Burch (2008) improved this method by using syntactic constraints and multiple languages in parallel. We have implemented a combination of Bannard and Callison-Burch (2005) and Callison-Burch (2008), in which we use PoS filters instead of syntactic constraints to compare our results with. More details can be found in the Section 5.

Apart from methods that use parallel corpora mono-lingual pattern-based methods have been used to find term variations. Fahmi (2009) acquired term variation for the medical domain using a two-step model. As a first step an initial list of synonyms are extracted using a method adapted from DIPRE (Brin, 99). During this step syntactic patterns guide the extraction of candidate terms in the same way as they will guide the extraction in this paper. This first step results in a list of candidate synonyms that are further filtered following a method described in Lin et al. (2003), which uses Web pages as an external source to measure the synonym compatibility hits of each pair. The precision and recall scores presented in Fahmi (2009) are high. We will give results for this method on our test set in Section 5 and refer to it as the pattern- and web-based approach.

3 Materials and methods

In the following subsections we describe the setup for our experiments.

3.1 Data collection

Measures of distributional similarity usually require large amounts of data. For the alignment method we need a parallel corpus of reasonable size with Dutch either as source or as target language coming from the domain we are interested in. Furthermore, we would like to experiment with various languages aligned to Dutch.

The freely available EMEA corpus (Tiedemann, 2009) includes 22 languages in parallel with a reasonable size of about 12-14 million tokens per language. The entire corpus is aligned at the sentence level for all possible combinations of languages. Thus, for acquiring Dutch synonyms we have 21 language pairs with Dutch as the source language. Each language pair includes about 1.1 million sentence pairs. Note that there is a lot of repetition in EMEA and the number of unique sentences (sentence fragments) is much smaller: around 350,000 sentence pairs per language pair with about 6-7 million tokens per language.

3.2 Word alignment and phrase extraction

For sentence alignment we applied *hunalign* (Varga et al., 2005) with the 'realign' function that induces lexical features from the bitext to be combined with length based features. Word alignment has been performed using GIZA++ (Och, 2003). We used standard settings defined in the Moses toolkit (Koehn et al., 2007) to generate Viterbi word alignments of IBM model 4 for sentences

not longer than 80 tokens. In order to improve the statistical alignment we used lowercased tokens and lemmas in case we had them available (produced by the *Tree-Tagger* (Schmid, 1994) and the Alpino parser (van Noord, 2006)).

We used the *grow* heuristics to combine the asymmetric word alignments which starts with the intersection of the two Viterbi alignments and adds block-neighboring points to it in a second step. In this way we obtain high precision links with some many-to-many alignments. Finally we used the phrase extraction tool from Moses to extract phrase correspondences. Phrases in statistical machine translation are defined as sequences of consecutive words and phrase extraction refers to the exhaustive extraction of all possible phrase pairs that are consistent with the underlying word alignment. Consistency in this case means that words in a legal phrase are only aligned to words in the corresponding phrase and not to any other word outside of that phrase. The extraction mechanism can be restricted by setting a maximum phrase length which is seven in the default settings of Moses. However, we set the maximum phrase length to four, because we do not expect many terms in the medical domain to be longer than 4 words.

As explained above, word alignment is carried out on lowercased and possibly lemmatised versions of the corpus. However, for phrase extraction, we used surface wordforms and extracted them along with the part-of-speech (PoS) tags for Dutch taken from the corresponding Alpino parse trees. This allows us to lowercase all words except the words that have been tagged as *name*. Furthermore, the inclusion of PoS tags enabled us to filter the resulting phrase table according to typical patterns of multi-word terms. We also removed phrases that consist of only non-alphabetical characters. Note that we rely entirely on automatic processing of our data. Thus, the results from automatic tagging, lemmatisation and word alignment include errors. Bannard and Callison-Burch (2005) show that when using manual alignment the percentage of correct paraphrases significantly rises from 48.9% to 74.9%.

3.3 Selecting candidate terms

As we explained above we can select those phrases that are more likely to be good terms by using a regular expression over PoS tags. We apply a pattern using adjectives (A), nouns (NN), names (NM) and prepositions (P) as its components based on Justeson and Katz. (1995) which was adapted to Dutch by Fahmi (2009):

```
((A|NN|NM)+|((A|NN|NM)*
(NN|NM P)?)(A|NN|NM)*))NN+
```

To explain this regular expression in words, a candidate term is either a sequence of adjectives and/or nouns and/or names, ending in a noun or name or it consists of two such strings, separated by a single preposition.

After applying the filters and removing all hapaxes we are left with 9.76 M co-occurrences of a Dutch (multi-word) term and a foreign translation.

3.4 Comparing vectors

To compare the vectors of the terms we need a similarity measures. We have chosen to describe the functions used in this paper using an extension of the notation used by Lin (1998), adapted by Curran (2003). Co-occurrence data is described as tuples: ⟨word, language, word′⟩, for example, ⟨kat, EN, cat⟩.

Asterisks indicate a set of values ranging over all existing values of that component of the relation tuple. For example, $(w, *, *)$ denotes for a given word w all translational contexts it has been found in in any language. For the example of *kat* in, this would denote all values for all translational contexts the word is found in: *Katze_DE*:17, *chat_FR*:26 etc. Everything is defined in terms of co-occurrence data with non-zero frequencies. The set of attributes or features for a given corpus is defined as:

$$(w, *, *) \equiv \{(r, w') | \exists (w, r, w')\}$$

Each pair yields a frequency value, and the sequence of values is a vector indexed by $r{:}w'$ values, rather than natural numbers. A subscripted asterisk indicates that the variables are bound together:

$$\sum (w_m, *_r, *_{w'}) \times (w_n, *_r, *_{w'})$$

The above refers to a dot product of the vectors for term w_m and term w_n summing over all the $r{:}w'$ pairs that these two terms have in common. For example we could compare the vectors for *kat* and some other term by applying the dot product to all bound variables.

We have limited our experiments to using Cosine[2]. We chose this measure, since it performed best in experiments reported in Van der Plas (2008). Cosine is a geometrical measure. It returns the cosine of the angle between the vectors of the words and is calculated as the dot product of the vectors:

$$Cosine = \frac{\sum(W1, *_r, *_{w'}) \times (W2, *_r, *_{w'})}{\sqrt{\sum(W1, *, *)^2 \times \sum(W2, *, *)^2}}$$

If the two words have the same distribution the angle between the vectors is zero.

3.5 Post-processing

A well-known problem of phrase-based methods to paraphrase or term variation acquisition is the fact that a large proportion of the term variations or paraphrases proposed by the system are super- or sub-strings of the original term (Callison-Burch, 2008). To remedy this problem we removed all term variations that are either super- or sub-strings of the original term from the lists of candidate term variations output by the system.

4 Evaluation

There are several evaluation methods available to assess lexico-semantic data. Curran (2003) distinguishes two types of evaluation: direct evaluation and indirect evaluation. Direct evaluation methods compare the semantic relations given by the system against human performance or expertise. Indirect approaches evaluate the system by measuring its performance on a specific task.

Since we are not aware of a task in which we could test the term variations for the Dutch medical domain and ad-hoc human judgments are time consuming and expensive, we decided to compare against a gold standard. Thereby denying the common knowledge that the drawback of using gold standard evaluations is the fact that gold standards often prove to be incomplete. In previous work on synonym acquisition for the general domain, Van der Plas and Tiedemann (2006) used the synsets in Dutch EuroWordnet (Vossen, 1998) for the evaluation of the proposed synonyms. In an evaluation with human judgments, Van der Plas and Tiedemann (2006) showed that in 37% of the cases the majority of the subjects judged the synonyms proposed by the system to be correct even though they were not found to be synonyms in Dutch EuroWordnet. For evaluating medical term variations in Dutch there are not many gold standards available. Moreover, the gold standards that are available are even less complete than for the general domain.

4.1 Gold standard

We have chosen to evaluate the nearest neighbours of the alignment-based method on the term variations from the Elseviers medical encyclopedia which is intended for the general audience containing 379K words. The encyclopedia was made available to us by Spectrum B.V.[3].

The test set is comprised of 848 medical terms from *aambeeld* 'incus' to *zwezerik* 'thymus' and their term variations. About 258 of these entries contain multiword terms. For most of the terms the list from Elseviers medical encyclopedia gives only one term variation, 146 terms have two term variations and only one term has three variations. For each of these medical terms in the test set the system generates a ranked list of term variations that will be evaluated against the term variations in the gold standard.

[2]Feature weights have been used in previous work for syntax-based methods to account for the fact that co-occurrences have different information values. Selectionally weak (Resnik, 1993) or *light* verbs such as *hebben* 'to have' have a lower information value than a verb such as *uitpersen* 'squeeze' that occurs less frequently. Although weights that promote features with a higher information value work very well for syntax-based methods, Van der Plas (2008) showed that weighting only helps to get better synonyms for very infrequent nouns when applied in alignment-based approaches. In the current setting we do not consider very infrequent terms so we did not use any weighting.

[3]http://www.kiesbeter.nl/medischeinformatie/

5 Results and Discussion

Before we present our results and give a detailed error analysis we would like to remind the reader of the two methods we compare our results with and give some more detail on the implementation of the second method.

5.1 Two methods for comparison

The first method is the pattern- and web-based approach described in Fahmi (2009). Note that we did not re-implement the method, so we were not able to run the method on the same corpus we are using in our experiments. The corpus used in Fahmi (2009) is a medical corpus developed in Tilburg University (http://ilk.uvt.nl/rolaquad). It consists of texts from a medical encyclopedia and a medical handbook and contains 57,004 sentences. The system outputs a ranked list of term variation pairs. We selected the top-100 pairs that are output by the system and evaluated these on the test set described in Subsection 4.1. The method is composed of two main steps. In the first step candidate terms are extracted from the corpus using a PoS filter, that is similar to the PoS filter we applied. In the second step pairs of candidate term variations are re-ranked on the basis of information from the Web. Phrasal patterns such as $XorY$ are used to get synonym compatibility hits as opposed to $XandY$ that points to non-synonymous terms.

The second method we compare with is the phrase-based translation method first introduced by Bannard and Callison-Burch (2005). Statistical word alignment can be used to measure the relation between source language items. Here, one makes use of the estimated translation likelihoods of phrases ($p(f|e)$ and $p(e|f)$) that are used to build translation models in standard phrase-based statistical machine translation systems (Koehn et al., 2007). Bannard and Callison-Burch (2005) define the problem of paraphrasing as the following search problem:

$$\hat{e_2} = argmax_{e_2:e_2 \neq e_1} p(e_2|e_1) \quad \text{where}$$

$$p(e_2|e_1) \approx \sum_f p(f|e_1)p(e_2|f)$$

Certainly, for paraphrasing we are not only interested in $\hat{e_2}$ but for the top-ranked paraphrase candidates but this essentially does not change the algorithm. In their paper, Bannard and Callison-Burch (2005) also show that systematic errors (usually originating from bad word alignments) can be reduced by summing over several language pairs.

$$\hat{e_2} \approx argmax_{e_2:e_2 \neq e_1} \sum_C \sum_{f_C} p(f_C|e_1)p(e_2|f_C)$$

This is the approach that we also adapted for our comparison. The only difference in our implementation is that we applied a PoS-filter to extract candidate terms as explained in section 3.3. In some sense this is a sort of syntactic constraint introduced in Callison-Burch (2008). Furthermore, we set the maximum phrase length to 4 and applied the same post-processing as described in Subsection 3.5 to obtain comparable results.

5.2 Results

Table 2 shows the results for our method compared with the method adapted from Bannard and Callison-Burch (2005) and the method by Fahmi (2009). Precision and recall are given at several values of k. At k=1, only the top-1 term variations the system proposes are taken into account. At k=3 the top-3 candidate term variations are included in the calculations.

The last column shows the coverage of the system. A coverage of 40% means that for 40% of the 850 terms in the test set one or more term variations are found. Recall is measured for the terms covered by the system.

From Table 2 we can read that the method we propose is able to get about 30% of the term variations right, when only the top-1 candidates are considered. It is able to retrieve roughly a quarter of the term variations provided in the gold standard[4]. If we increase k precision goes down and recall goes up. This is expected, because the system proposes a ranked list of candidate term variations so at higher values of k the quality is lower, but more terms from the gold standard are found.

[4]Note that a recall of 100% is not possible, because some terms have several term variations.

Method	k=1		k=2		k=3		Coverage
	P	R	P	R	P	R	
Phrase-based Distr. Sim	28.9	22.8	21.8	32.7	17.3	37.2	40.0
Bannard&Callison-Burch (2005)	18.4	15.3	16.9	27.3	13.7	32.3	48.1
Fahmi (2009)	38.2	35.1	37.1	35.1	37.1	35.1	4.0
Phrase-based Distr. Sim (hapaxes)	25.4	20.9	20.4	32.1	16.1	36.8	47.8

Table 2: Percent precision and recall at several values of k and percent coverage for the method proposed in this paper (plus a version including hapaxes), the method adapted from Bannard and Callison-Burch (2005) and the output of the system proposed by Fahmi (2009)

In comparison, the scores resulting from our adapted implementation of Bannard and Callison-Burch (2005) are lower. They do however, manage to find more terms from the test set covering around 48% of the words in the gold standard. This is due to the cut-off that we use when creating the co-occurrence vector to remove unreliable data points. In our approach we discarded hapaxes, whereas for the Bannard and Callison-Burch approach the entire phrase table is used. We therefore ran our system once again without this cut-off. As expected, the coverage went up in that setting – actually to 48% as well.[5] However, we can see that the precision and recall remained higher, than the scores we got with the implementation following Bannard and Callison-Burch (2005). Hence, our vector-based approach seems to outperform the direct use of probabilities from phrase-based MT.

Finally, we also compare our results with the data set extracted using the pattern- and web-based approach from Fahmi (2009). The precision and recall figures of that data set are the highest in our comparison. However, since the coverage of this method is very low (which is not surprising since a smaller corpus is used to get these results) the precision and recall are calculated on the basis of a very small number of examples (35 to be precise). The results are therefore not very reliable. The precision and recall figures presented in Fahmi (2009), however, point in the same direction. To get an idea of the actual coverage of this method we would need to apply this extraction technique to the EMEA corpus. This is especially difficult due to the heavy use of web queries which makes it problematic to apply this method to large data sets.

[5]The small difference in coverage is due to some mistakes in tokenisation for our method.

5.3 Error analysis

The most important finding we did, when closely inspecting the output of the system is that many of the term variations proposed by the system are not found in the gold standard, but are in fact correct. Here, we give some examples below:

arts, dokter (‘doctor’)
ademnood, ademhalingsnood (‘respiratory distress’)
aangezichtsverlamming, gelaatsparalyse (‘facial paralysis’)
alvleesklierkanker, pancreaskanker (‘cancer of the pancreas’)

The scores given in Table 2 are therefore pessimistic and a manual evaluation with domain specialist would certainly give us more realistic and probably much higher scores. We also found some spelling variants which are usually not covered by the gold standard. Look, for instance, at the following examples:

astma, asthma (‘asthma’)
atherosclerose, Artherosclerosis (‘atherosclerosis’)
autonoom zenuwstelsel, autonome zenuwstelsel (‘autonomic nervous system’)

Some mistakes could have been avoided using stemming or proper lemmatisation (plurals that are counted as wrong):

abortus, zwangerschapsafbrekingen (‘abortion’)
adenoom, adenomen (‘adenoma’)
indigestie, spijsverteringsstoornissen (‘indigestion’)

After removing the previous cases from the data, some of the remaining mistakes are related to the problem we mentioned in section 3.5: Phrase-

based methods to paraphrase or term variation acquisition have the tendency to propose term variations that are super- or sub-strings of the original term. We were able to filter out these super- or sub-strings, but not in cases where a candidate term is a term variation of a super- or sub-string of the original term. Consider, for example the term *bloeddrukverlaging* 'blood pressure decrease' and the candidate *afname* 'decrease', where *afname* is a synonym for *verlaging*.

6 Conclusions

In this article we have shown that translational context together with measures of distributional similarity can be used to extract medical term variations from aligned parallel corpora. Automatic word alignment and phrase extraction techniques from statistical machine translation can be applied to collect translational variations across various languages which are then used to identify semantically related words and phrases. In this study, we additionally apply pattern-based filters using part-of-speech labels to focus on particular patterns of single and multi-word terms. Our method outperforms another alignment-based approach measured on a gold standard taken from a medical encyclopedia when applied to the same data set and using the same PoS filter. Precision and recall are still quite poor according to the automatic evaluation. However, manual inspection suggests that many candidates are simply misjudged because of the low coverage of the gold standard data. We are currently setting up a manual evaluation. Altogether our approach provides a promising strategy for the extraction of term variations using straightforward and fully automatic techniques. We believe that our results could be useful for a range of applications and resources and that the approach in general is robust and flexible enough to be applied to various languages and domains.

Acknowledgements

The research leading to these results has received funding from the EU FP7 programme (FP7/2007-2013) under grant agreement nr 216594 (CLASSIC project: www.classic-project.org).

References

Bannard, C. and C. Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the annual Meeting of the Association for Computational Linguistics (ACL)*.

Brin, S. 99. Extracting patterns and relations from the World Wide Web. In *WebDB '98: Selected papers from the International Workshop on The World Wide Web and Databases*.

Brown, P.F., S.A. Della Pietra, V.J. Della Pietra, and R.L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–296.

Callison-Burch, C. 2008. Syntactic constraints on paraphrases extracted from parallel corpora. In *Proceedings of EMNLP*.

Curran, J.R. 2003. *From Distributional to Semantic Similarity*. Ph.D. thesis, University of Edinburgh.

Dyvik, H. 1998. Translations as semantic mirrors. In *Proceedings of Workshop Multilinguality in the Lexicon II (ECAI)*.

Dyvik, H. 2002. Translations as semantic mirrors: from parallel corpus to wordnet. *Language and Computers, Advances in Corpus Linguistics. Papers from the 23rd International Conference on English Language Research on Computerized Corpora (ICAME 23)*, 16:311–326.

Fahmi, I. 2009. *Automatic Term and Relation Extraction for Medical Question Answering System*. Ph.D. thesis, University of Groningen.

Firth, J.R. 1957. A synopsis of linguistic theory 1930-1955. *Studies in Linguistic Analysis (special volume of the Philological Society)*, pages 1–32.

Harris, Z.S. 1968. *Mathematical structures of language*. Wiley.

Ide, N., T. Erjavec, and D. Tufis. 2002. Sense discrimination with parallel corpora. In *Proceedings of the ACL Workshop on Sense Disambiguation: Recent Successes and Future Directions.*

Justeson, J. and S. Katz. 1995. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1:9–27.

Koehn, P., H. Hoang, A. Birch, C. Callison-Burch, M.Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A.Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Koehn, P. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the MT Summit*, pages 79–86, Phuket, Thailand.

Lin, D., S. Zhao, L. Qin, and M. Zhou. 2003. Identifying synonyms among distributionally similar words. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Lin, D. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of COLING/ACL*.

McCray, A. and W. Hole. 1990. The scope and structure of the first version of the umls semantic network. In *Symposium on Computer Applications in Primary Care (SCAMC-90)*, IEEE Computer Society, pages 126–130, , Washington DC, IEEE Computer Society. 126-130.

Och, F.J. 2003. GIZA++: Training of statistical translation models. Available from `http://www.isi.edu/~och/GIZA++.html`.

Resnik, P. and D. Yarowsky. 1997. A perspective on word sense disambiguation methods and their evaluation. In *Proceedings of ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, what, and how?*

Resnik, P. 1993. Selection and information. Unpublished doctoral thesis, University of Pennsylvania.

Schmid, Helmut. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK, September. http://www.ims.uni-stuttgart.de/~schmid/.

Schütze, H. 1992. Dimensions of meaning. In *Proceedings of the ACM/IEEE conference on Supercomputing*.

Shimota, M. and E. Sumita. 2002. Automatic paraphrasing based on parallel corpus for normalization. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*.

Tiedemann, Jörg. 2009. News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In Nicolov, N., K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing*, volume V, pages 237–248, Borovets, Bulgaria. John Benjamins, Amsterdam/Philadelphia.

van der Plas, L. and J. Tiedemann. 2006. Finding synonyms using automatic word alignment and measures of distributional similarity. In *Proceedings of COLING/ACL*.

van der Plas. 2008. *Automatic lexico-semantic acquisition for question answering*. Groningen dissertations in linguistics.

van Noord, G. 2006. At last parsing is now operational. In *Actes de la 13eme Conference sur le Traitement Automatique des Langues Naturelles*.

Varga, D., L. Nmeth, P. Halcsy, A. Kornai, V. Trn, and V. Nagy. 2005. Parallel corpora for medium density languages. In *Proceedings of RANLP 2005*, pages 590–596.

Vossen, P. 1998. EuroWordNet a multilingual database with lexical semantic networks.

Wilks, Y., D. Fass, Ch. M. Guo, J. E. McDonald, and B. M. Slator T. Plate. 1993. Providing machine tractable dictionary tools. *Machine Translation*, 5(2):99–154.

Wu, H. and M. Zhou. 2003. Optimizing synonym extraction using monolingual and bilingual resources. In *Proceedings of the International Workshop on Paraphrasing: Paraphrase Acquisition and Applications (IWP)*.

Learning Semantic Network Patterns for Hypernymy Extraction

Tim vor der Brück
Intelligent Information and Communication Systems (IICS)
FernUniversität in Hagen
tim.vorderbrueck@fernuni-hagen.de

Abstract

Current approaches of hypernymy acquisition are mostly based on syntactic or surface representations and extract hypernymy relations between surface word forms and not word readings. In this paper we present a purely semantic approach for hypernymy extraction based on semantic networks (SNs). This approach employs a set of patterns
$\text{SUB0}(a1, a2) \leftarrow$ *premise* where the premise part of a pattern is given by a SN. Furthermore this paper describes how the patterns can be derived by relational statistical learning following the Minimum Description Length principle (MDL). The evaluation demonstrates the usefulness of the learned patterns and also of the entire hypernymy extraction system.

1 Introduction

A concept is a hypernym of another concept if the first concept denotes a superset of the second. For instance, the class of *animals* is a superset of the class of *dogs*. Thus, animal is a hypernym of its hyponym dog and a hypernymy relation holds between animal and dog. A large collection of hypernymy (supertype) relations is needed for a multitude of tasks in natural language processing. Hypernyms are required for deriving inferences in question answering systems, they can be employed to identify similar words for information retrieval or they can be useful to avoid word-repetition in natural language generation systems. To build a taxonomy manually requires a large amount of work. Thus, automatic approaches for their construction are preferable. In this work we introduce a semantically oriented approach where the hypernyms are extracted using a set of patterns which are neither syntactic nor surface-oriented but instead purely semantic and are based on a SN formalism. The patterns are applied on a set of SNs which are automatically derived from the German Wikipedia[1] by a deep syntactico-semantic analysis. Furthermore, these patterns are automatically created by a machine learning approach based on the MDL principle.

2 Related Work

Patterns for hypernymy extraction were first introduced by Hearst (Hearst, 1992), the so-called Hearst patterns. An example of such a pattern is:
NP_{hypo} {,NP_{hypo}}*{,} and other NP_{hyper}.

These patterns are applied on arbitrary texts and the instantiated variables NP_{hypo} and NP_{hyper} are then extracted as a concrete hypernymy relation.

Apart from the handcrafted patterns there was also some work to determine patterns automatically from texts (Snow and others, 2005). For that, Snow et al. collected sentences in a given text corpus with known hypernym noun pairs. These sentences are then parsed by a dependency parser. Afterwards, the path in the dependency tree is extracted which connects the corresponding nouns with each other. To account for certain key words indicating a hypernymy relation like *such* (see first Hearst pattern) they added the links to the word on either side of the two nouns (if not yet contained) to the path too. Frequently oc-

[1] Note that for better readability the examples are translated from German into English throughout this paper.

Proceedings of the 6th Workshop on Ontologies and Lexical Resources (Ontolex 2010), pages 38–47,
Beijing, August 2010

curring paths are then learned as patterns for indicating a hypernymy relation.

An alternative approach for learning patterns which is based on a surface instead of a syntactic representation was proposed by Morin et al. (Morin and Jaquemin, 2004). They investigate sentences containing pairs of known hypernyms and hyponyms as well. All these sentences are converted into so-called "lexico-syntactic expressions" where all NPs and lists of NPs are replaced by special symbols, e.g.: *NP find in NP such as LIST*. A similarity measure between two such expressions is defined as the sum of the maximal length of common substrings for the maximum text windows before, between and after the hyponym/hypernym pair. All sentences are then clustered according to this similarity measure. The representative pattern (called *candidate pattern*) of each cluster is defined to be the expression with the lowest mean square error (deviation) to all other expressions in the same similarity cluster. The patterns to be used for hyponymy detection are the candidate patterns of all clusters found.

3 MultiNet

MultiNet is an SN formalism (Helbig, 2006). In contrast to SNs like WordNet (Fellbaum, 1998) or GermaNet (Hamp and Feldweg, 1997), which contain lexical relations between synsets, MultiNet is designed to comprehensively represent the semantics of natural language expressions. An SN in the MultiNet formalism is given as a set of vertices and arcs where the vertices represent the concepts (word readings) and the arcs the relations (or functions) between the concepts. A vertex can be lexicalized if it is directly associated to a lexical entry or non-lexicalized. An example SN is shown in Fig. 1. Note that each vertex of the SN is assigned both a unique ID (e.g., *c2*) and a label which is the associated lexical entry for lexicalized vertices and *anon* for non-lexicalized vertices. Thus, two SNs differing only by the IDs of the non-lexicalized vertices are considered equivalent. Important MultiNet relations/functions are (Helbig, 2006):

- AGT: Conceptual role: Agent
- ATTR: Specification of an attribute
- VAL: Relation between a specific attribute and its value
- PROP: Relation between object and property
- *ITMS: Function enumerating a set
- PRED: Predicative concept characterizing a plurality
- OBJ: Neutral object
- SUB0: Relation of conceptual subordination (hyponymy) and hyperrelation to SUBR, SUBS, and SUB
- SUBS: Relation of conceptual subordination (for situations)
- SUBR: Relation of conceptual subordination (for relations)
- SUB: Relation of conceptual subordination other than SUBS and SUBR

MultiNet is supported by a semantic lexicon (Hartrumpf and others, 2003) which defines, in addition to traditional grammatical entries like gender and number, semantic information consisting of one or more ontological sorts and several semantic features for each lexicon entry. The ontological sorts (more than 40) form a taxonomy. In contrast to other taxonomies, ontological sorts are not necessarily lexicalized, i.e., they need not denote lexical entries. The following list shows a small selection of ontological sorts which are inherited from *object*:

- Concrete objects: e.g., *milk*, *honey*
  - Discrete objects: e.g., *chair*
  - Substances: e.g.,, *milk*, *honey*
- Abstract objects: e.g., *race*, *robbery*

Semantic features denote certain semantic properties for objects. Such a property can either be present, not present or underspecified. A selection of several semantic features is given below:

ANIMAL, ANIMATE, ARTIF (artificial), HUMAN, SPATIAL, THCONC (theoretical concept)

Example for the concept *bottle.1.1*[2]: discrete object; ANIMAL -, ANIMATE -, ARTIF +, HUMAN -, SPATIAL +, THCONC -, ...

[2]the suffix *.1.1* denotes the reading numbered *.1.1* of the word *bottle*.

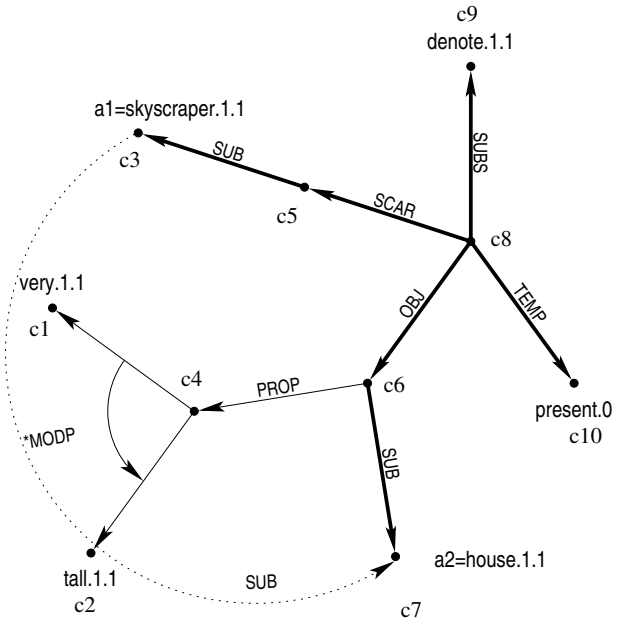

Figure 1: Matching a pattern to an SN. Bold lines indicate matched arcs, the dashed line the inferred arc.

The SNs as described here are automatically constructed from (German) texts by the deep linguistic parser WOCADI[3](Hartrumpf, 2002) whose parsing process is based on a word class functional analysis.

4 Application of Deep Patterns

The extraction of hyponyms as described here is based on a set of patterns. Each pattern consists of a conclusion part SUB0($a1$, $a2$) and a premise part in form of an SN where both *a1* and *a2* have to show up. The patterns are applied by a pattern matcher (or automated theorem prover if axioms are used) which matches the premise with an SN. The variable bindings for *a1* and *a2* are given by the matched concepts of the SN. An example pattern which matches to the sentence: *A skyscraper denotes a very tall building.* is D_4 (see Table 1). The pattern matching process is illustrated in Fig.1. The resulting instantiated conclusion which is stored in the knowledge base is SUB0($skyscraper$.1.1, $house$.1.1). Advantages by using the MultiNet SN formalism for hypernym (and instance-of relation) acquisition consists of: learning relations between word readings instead of words, the possibility to apply logical axioms and background knowledge, and that person names are already parsed.

An example sentence from the Wikipedia corpus where a hypernymy relation was successfully extracted by our deep approach and which illustrates the usefulness of this approach is: *In any case, not all incidents from the Bermuda Triangle or from other world areas are fully explained.* From this sentence, a hypernymy pair cannot be extracted by the Hearst pattern *X or other Y*. The application of this pattern fails due to the word *from* which cannot be matched. To extract this relation by means of shallow patterns an additional pattern would have to be introduced. This could also be the case if syntactic patterns were used instead since the coordination of *Bermuda Triangle* and *world areas* is not represented in the syntactic constituency tree but only on a semantic level[4].

[3]WOCADI is the abbreviation for **wo**rd **cla**ss **di**sambiguation.

[4]Note that some dependency parsers normalize some syntactic variations too.

5 Graph Substructure Learning By Following the Minimum Description Length Principle

In this section, we describe how the patterns can be learned by a supervised machine learning approach following the Minimum Description Length principle. This principle states that the best hypothesis for a given data set is that one which minimizes the description of the data (Rissanen, 1989), i.e., compresses the data the most. Basically we follow the substructure learning approach of Cook and Holder (Cook and Holder, 1994).

According to this approach, the description length to minimize is the number of bits required to encode a certain graph which is compressed by means of a substructure. If a lot of graph vertices can be matched with the substructure vertices, this description length will be quite small. For our learning scenario we investigate collection of SNs containing a known hypernymy relationship. A pattern (given by a substructure in the premise) which compresses this set quite well is expected to be useful for extracting hypernyms.

Let us first determine the number of bits to encode the entire graph or SN. A graph can be represented by its adjacency matrix and a set of vertex and arc labels. Since an adjacency matrix consists only of ones and zeros, it is well suitable for a binary encoding. For the encoding process, we do not regard the label names directly but instead their number assuming an ordering exists on the label names (e.g., alphabetical).

	$c1$	$c2$	$c3$	$c4$	$c5$	$c6$	$c7$	$c8$	$c9$	$c10$
$c1$	0	0	0	0	0	0	0	0	0	0
$c2$	0	0	0	0	0	0	0	0	0	0
$c3$	0	0	0	0	0	0	0	0	0	0
$c4$	1	1	0	0	0	0	0	0	0	0
$c5$	0	0	1	0	0	0	0	0	0	0
$c6$	0	0	0	0	0	0	1	0	0	0
$c7$	0	0	0	0	0	0	0	0	0	0
$c8$	0	0	0	0	1	1	0	0	1	1
$c9$	0	0	0	0	0	0	0	0	0	0
$c10$	0	0	0	0	0	0	0	0	0	0

Figure 2: Adjacency matrix of the SN.

To encode all labels the number of labels and a list of all label numbers have to be specified, e.g., *3,1,2,1* for 3 vertices with two different label numbers[5] (1,2). The first number encoding (3) starts at position 0 in the bit string, the second (1) at position $2 = \lceil \log_2 3 \rceil$, the third one at position $2 + \lceil \log_2 2 \rceil$, etc. Since the graph actually need not to be encoded in this way but only the length of the encoding is important, non-integer numbers of bits are accepted for simplicity too. If there are a total of l_u different labels, then each encoded label number requires $\log_2(l_u)$ bits. The total number of bits to encode the vertex labels are then given by:
$vbits = \log_2(v) + v \log_2(l_u)$ in which v denotes the total number of vertices[6].

In the next step, the adjacency matrix is encoded where each row is processed separately. A straightforward approach for encoding one row would be to use v number of bits, one for every column. However, the number of zeros are generally much larger than the number of ones which means that a better compression of the data is possible by exploiting this fact. Consider the case that a certain matrix row contains exactly m ones. There are $\begin{pmatrix} v \\ m \end{pmatrix}$ possibilities to distribute the ones to the individual cells. All possible permutations could be specified in a list. In this case it is only necessary to specify the position in this list to uniquely describe one row. Let $b = \max_i k_i$. Then the number of ones in one row can be encoded using $\log_2(b+1)$ bits. $\log_2 \begin{pmatrix} v \\ k_i \end{pmatrix}$ bits are required to encode the distribution of ones in one row. Additionally, $\log_2(b+1)$ bits are needed to encode b which is only necessary once for the matrix. Let us consider the adjacency matrix given in Fig. 2 of the SN shown in Fig. 1 with 10 rows and columns where each row contains at most four ones. To encode the row c_4, containing two ones, re-

[5] The commas are only included for better readability and are actually not encoded.

[6] The approach of Cook and Holder is a bit inexact here. To be precise, the number of bits needed to encode v and b would have to be known a priori.

quires $\log_2(4) + \log_2 \binom{10}{2} = 7.49$ bits which is smaller than 10 bits which were necessary for the naïve approach. The total length *rbits* of the encoding is given by:

$$\begin{aligned} rbits =& \log_2(b+1) + \sum_{i=1}^{v} [\log_2(b+1) + \\ & \log_2 \binom{v}{k_i}] \\ =& (v+1)\log_2(b+1) + \\ & \sum_{i=1}^{v} \log_2 \binom{v}{k_i} \end{aligned} \tag{1}$$

Finally, the arcs need to be encoded. Let $e(i,j)$ be the number of arcs between vertex i and j in the graph and $m := max_{i,j} e(i,j)$. $\log_2(m)$ bits are required to encode the number of arcs between both vertices and $\log_2(l_e)$ bits are needed for the arc label (out of a set of l_e elements). Then the entire number of bits is given by (e is the number of arcs in the graph):

$$\begin{aligned} ebits =& \log_2(m) + \sum_{i=1}^{v}\sum_{j=1}^{v} [A[i,j] log_2(m) + \\ & e(i,j)\log_2(l_e)] \\ =& \log_2(m) + e\log_2(l_e) + \\ & \sum_{i=1}^{v}\sum_{j=1}^{v} A[i,j]\log_2(m) \\ =& e(\log_2(l_e)) + (K+1)\log_2(m) \end{aligned} \tag{2}$$

where K is the number of ones in the adjacency matrix.

The total description length of the graph is then given by: $vbits + rbits + ebits$.

Now let us investigate how the description length of the compressed graph is determined. In the original algorithm the substructure is replaced in the graph by a single vertex. The description length of the graph compressed by the substructure is then given by the description length of the substructure added by the description length of the modified graph.

	$c1$	$c2$	$c3$	$c4$	$c5$	$c6$	$c7$	$c8$	$c9$	$c10$
$c1$	0	0	0	0	0	0	0	0	0	0
$c2$	0	0	0	0	0	0	0	0	0	0
$c3$	0	0	0	0	0	0	0	0	0	0
$c4$	1	1	0	0	0	0	0	0	0	0
$c5$	0	0	×	0	0	0	0	0	0	0
$c6$	0	0	0	0	0	0	×	0	0	0
$c7$	0	0	0	0	0	0	0	0	0	0
$c8$	0	0	0	0	×	×	0	0	×	×
$c9$	0	0	0	0	0	0	0	0	0	0
$c10$	0	0	0	0	0	0	0	0	0	0

Figure 3: Adjacency matrix of the compressed SN. Vertices whose connections can be completely inferred from the pattern are removed.

In our method there are two major differences from the graph learning approach of Cook and Holder.

- Not a single graph is compressed but a set of graphs.
- For the approach of Cook and Holder, it is unknown which vertex of the substructure a graph node is actually connected with. Thus, the description is not complete and the original graph could not be reconstructed using the substructure and the compressed graph. To make the description complete we specify the bindings of the substructure vertices to the graph vertices.

The generalization of the Cook and Holder-algorithm to a set of graphs is quite straight forward. The total description length of a set of compressed graphs is given by the description length of the substructure (here pattern) added to the sum of the description lengths of each SN compressed by this pattern.

Additional bits are needed to encode the vertex bindings (assuming the pattern premise is contained in the SN). First the number of bindings *bin* ($[1, v_p]$, v_p: number of non-lexicalized vertices appearing in a pattern) has to be specified which requires $\log_2(v_p)$ bits. The number of bits needed to encode a single binding is given by $\log_2(v_p) + \log_2(v)$ (vertex indices: $[0, v_p - 1]$ to $[0, v-1]$). Thus, the total

number of required bits is given by

$$binbits = bin(\log_2(v_p) + \log_2(v)) + \log_2(v_p) \quad (3)$$

Note that not all bindings need to be encoded. The number of required binding encodings can be determined as follows. First all bindings for all non-lexicalized pattern vertices are determined. Then all cells from the adjacency matrix of the SN which contain a one and are also contained in the adjacency matrix of the pattern, if this binding is applied to the non-lexicalized pattern vertices, are set to zero. Vertices which contain only zeros in the adjacency matrix on both columns and rows are removed from the adjacency matrix/graph. The arcs from and to this vertex can be completely inferred by the pattern which means that all vertices this vertex is connected with are also contained in the pattern. Since SNs differing only by the IDs of their non-lexicalized vertices are considered identical, no binding has to be specified for such a vertex. Additionally, the modified adjacency matrix is the result of the compression by the pattern, i.e., vbits, rbits, and ebits are determined from the modified adjacency matrix/graph if the pattern was successfully matched to the SN.

Let us consider our example pattern D_4 (Table 1). The following bindings are determined: a1: c3 (a1); a: c8; c: c6; b: c5; a2: c7 (a2)

The bindings for *a1* and *a2* need not to be remembered since all hyponym vertices are renamed to *a1* and the hypernym vertices to *a2* in order to learn generic patterns for arbitrary hypernyms/hyponyms. The cells of the adjacency matrix which are associated to the arcs: $\text{SCAR}(c8, c5)$, $\text{SUB}(c5, a1)$, $\text{OBJ}(c8, c6)$, $\text{SUBS}(c8, c9)$, $\text{TEMP}(c8, c10)$ are set to zero (marked by a cross in Fig. 3) since these arcs are also represented in the pattern using the bindings stated above. The rows and columns of $c3$, $c5$, $c7$, and $c9$ of the modified graph adjacency matrix only contain zeros. Thus, these rows can be removed from the adjacency matrix and the associated concepts can be eliminated from the vertex set of the SN.

The findings of the optimal patterns is done compositionally employing a beam search approach. First this approach starts with patterns containing only a single arc. These patterns are then extended by adding one arc after another preferring patterns leading to small description lengths of the compressed SNs. Note that only pattern premises are allowed which are fully connected, e.g., $\text{SUB}(a, c) \wedge \text{SUB}(e, f)$ is no acceptable premise.

Two lists are used during the search, $local\_best_i$ for guiding the search process and *global_best* for storing the best global results found so far:

- $local\_best_i$: The k best patterns of length i
- *global_best*: The k best patterns of any length

The list $local\_best_i$ is determined by extending all elements from $local\_best_{i-1}$ by one arc and only keeping the k arcs leading to the smallest description length. The list *global_best* is updated after each change of the list $local\_best_i$. This process is iterated as long as the total description length can be further reduced, i.e., $DL(local\_best_{i+1}[0]) < DL(local\_best_i[0])$, where $DL : Pattern \rightarrow \mathbb{R}$ denotes the description length of a pattern and [0] accesses the first element of a list.

The list *global_best* contains as the result of this approach the k patterns with the smallest overall compressed description length[7]. Note however that it is often not recommended to use all elements of *global_best* since this list contains oftentimes patterns where the premise part is a subgraph (can be inferred by) another premise pattern part contained in this list and their combination would actually not reduce the description length. Thus, in addition to the original approach of Cook and Holder, a dependency resolution is done.

The following iterative approach is proposed to cancel out such dependent patterns:

1. Start with the first entry of the global list: $depend\_best := \{global\_best[0]\}$

[7] compressed description length: short for description length of the SNs compressed by the pattern

ID	Definition	Matching Expression
D_1	$\text{SUB}0(a1, a2) \leftarrow$ $\text{SUB}(g, a2) \wedge \text{ATTCH}(g, f) \wedge$ $\text{SUBR}(e, sub.0) \wedge \text{TEMP}(e, present.0) \wedge$ $\text{ARG}2(e, f) \wedge \text{ARG}1(e, d) \wedge$ $\text{SUB}(d, a1)$	An apple$_{hypo}$ is a type of fruit$_{hyper}$.
D_2	$\text{SUB}0(a1, a2) \leftarrow$ $\text{SUB}(f, a2) \wedge \text{EQU}(g, f) \wedge$ $\text{SUBR}(e, equ.0) \wedge \text{TEMP}(e, present.0) \wedge$ $\text{ARG}2(e, f) \wedge \text{ARG}1(e, d) \wedge$ $\text{SUB}(d, a1)$	Psycho-linguistics$_{hypo}$ is a science$_{hyper}$ of the human ability to speak.
D_3	$\text{SUB}0(a1, a2) \leftarrow$ $\text{PRED}(g, a2) \wedge \text{ATTCH}(g, f) \wedge$ $\text{SUBR}(e, pred.0) \wedge \text{ARG}2(e, f) \wedge$ $\text{TEMP}(e, present.0) \wedge \text{ARG}1(e, d) \wedge$ $\text{PRED}(d, a1)$	Hepialidae$_{hypo}$ are a kind of insects$_{hyper}$. literal translation from: *Die Wurzelbohrer sind eine Familie der Schmetterlinge.*
D_4	$\text{SUB}0(a1, a2) \leftarrow$ $\text{SUB}(f, a2) \wedge \text{SUBS}(e, denote.1.1) \wedge$ $\text{TEMP}(e, present.0) \wedge \text{OBJ}(e, f) \wedge$ $\text{SCAR}(e, d) \wedge \text{SUB}(d, a1)$	A skyscraper$_{hypo}$ denotes a very tall building$_{hyper}$.
D_5	$\text{SUB}0(a1, a2) \leftarrow$ $\text{PROP}(f, other.1.1) \wedge \text{PRED}(f, a2) \wedge$ $foll_{*_{\text{ITMS}}}(d, f) \wedge \text{PRED}(d, a1)$	ducks$_{hypo}$ and other animals$_{hyper}$
D_6	$\text{SUB}0(a1, a2) \leftarrow$ $\text{SUB}(d, a2) \wedge \text{SUB}(d, a1)$	the instrument$_{hyper}$ cello$_{hypo}$
D_7	$\text{SUB}0(a1, a2) \leftarrow \text{SUB}(f, a2) \wedge$ $\text{TEMP}(e, present.0) \wedge \text{SUBR}(e, sub.0) \wedge$ $\text{SUB}(d, a1) \wedge \text{ARG}2(e, f) \wedge$ $\text{ARG}1(e, d)$	The Morton number$_{hypo}$ is a dimensionless indicator$_{hyper}$.

Table 1: A selection of automatically learned patterns.

2. Set *index*:=1
3. Calculate the combined (compressed) description length of *depend_best* and $\{global\_best[index]\}$
4. If the combined description length is reduced add $global\_best[index]$ to *depend_best*, otherwise leave *depend_best* unchanged
5. If $counter \geq length(global\_best)$ then return *depend_best*
6. $index := index + 1$
7. Go back to step 3

6 System Architecture

In this section, we give an overview over our hypernymy extraction system. The following procedure is employed to identify hypernymy relations in Wikipedia (see Fig. 4):

1. At first, all sentences of Wikipedia are analyzed by the deep analyzer WOCADI (Hartrumpf, 2002). As a result of the parsing process, a token list, a syntactic dependency tree, and an SN is created.

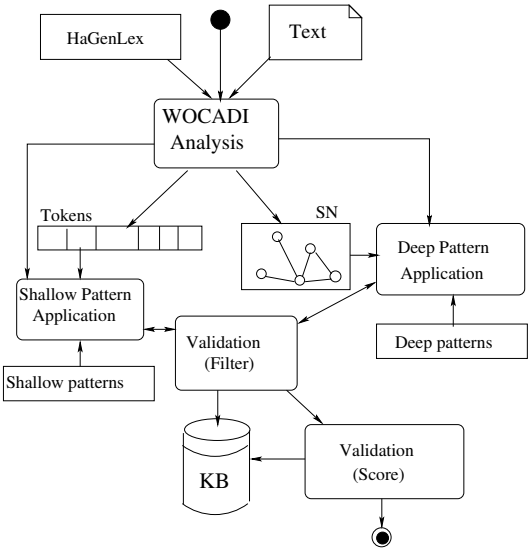

Figure 4: Activity diagram of the hypernym extraction process.

2. Shallow patterns based on regular expressions are applied to the token lists, and deep patterns (learned and hand-crafted) are applied to the SNs to generate proposals for hypernymy relations.
3. A validation tool using ontological sorts and semantic features checks whether the proposals are technically admissible at all to reduce the amount of data to be stored in the knowledge base KB.
4. If the validation is successful, the hypernymy hypothesis is integrated into KB. Steps 2–4 are repeated until all sentences are processed.
5. Each hypernymy hypothesis in KB is assigned a confidence score estimating its reliability.

7 Validation Features

The knowledge acquisition carried out is followed by a two-step validation. In the first step, we check the ontological sorts and semantic features of relational arguments for subsumption. For instance, a discrete concept (ontological sort: d) denoting a human being (semantic feature: human +) can only be hypernym of an other object, if this object is both discrete and a human being as well. Only relational candidates for which semantic features and ontological sorts can be shown to be compatible are stored in the knowledge base.

In a second step, each relational candidate in the knowledge base is assigned a quality score. This is done by means of a support vector machine (SVM) on several features. The SVM determines the classification (hypernymy or non-hypernymy) and a probability value for each hypernymy hypothesis. If the classification is 'hypernymy', the score is defined by this probability value, otherwise as one minus this value.

Correctness Rate: The feature *Correctness Rate* takes into account that the assumed hypernym alone is already a strong indication for the correctness or incorrectness of the investigated relation. The same holds for the assumed hyponym as well. For instance, relation hypotheses with hypernym *liquid* and *town* are usually correct. However, this is not the case for abstract concepts. Moreover, movie names are often extracted incompletely since they can consist of several tokens. Thus, this indicator determines how often a concept pair is classified correctly if a certain concept shows up in the first (hyponym) or second (hypernym) position.

Frequency: The feature *frequency* regards the quotient of the occurrences of the hyponym in other extracted relations in hyponym position and the hypernym in hypernym position.

This feature is based on two assumption. First, we assume that general terms normally occur more frequently in large text corpora than very specific ones (Joho and Sanderson, 2007). Second, we assume that usually a hypernym has more hyponyms than vice-versa.

Context: Generally, the hyponym can appear in the same textual context as its hypernym. The textual context can be described as a set of other concepts (or words for shallow approaches) which occur in the neighborhood of the investigated hyponym/hypernym candidate pair investigated on a large text corpus. Instead of the textual context we regard the semantic context. More specifically, the distributions of all concepts are regarded which are connected with the assumed hypernym/hyponym concept by the MultiNet-PROP (property) relation. The formula to estimate the similarity was basically taken from (Cimiano and others, 2005).

ID	Precision	First Sent.	# Matches
D_1	0.275	0.323	5 484
D_2	0.183	0.230	35 497
D_3	0.514	0.780	937
D_4	0.536	0.706	1 581
D_5	0.592	-	3 461
D_6	0.171	0.167	37 655

Table 2: Precision of hypernymy hypotheses extracted by patterns without usage of the validation component (D_7 not yet evaluated).

See (vor der Brück, 2010) for a more de-

Score	≥0.95	≥0.90	≥0.85	≥0.80	≥0.75	≥0.70	≥0.65	≥0.60	≥0.55
Precision	1.0000	0.8723	0.8649	0.8248	0.8203	0.7049	0.6781	0.5741	0.5703

Table 3: Precision of the extracted hypernymy relations for different confidence score intervals.

tailed description of the validation features.

8 Evaluation

We applied the pattern learning process on a collection of 600 SN, derived by WOCADI from Wikipedia, which contain hyponymically related concepts. Table 1 contains some of the extracted patterns including a typical expression to which this pattern could be matched. The predicate $foll_f(a, b)$ used in this table specifies that argument a precedes argument b in the argument list of function f. Patterns D_1-D_4 and D_7 contain concept definitions where the defined concept is, in many cases, the hyponym of the defining concept. In pattern D_1 and D_7 the defining concept is directly identified by the parser as hypernym of the defined concept (SUBR(e, *sub.0*)). In pattern D_2 the defining concept is recognized as equivalent to the defined concept (SUBR(e, *equ.0*)). However, in most of the cases the defining concept consists of a meaning molecule, i.e., a complex concept where some inner concept is modified by an additional expression (often a property or an additional subclause). If this expression is dropped which is done by the pattern D_2 the remaining concept becomes a hypernym of the defined concept. Pattern D_5 is a well-known Hearst pattern. Pattern D_6 is used to match to appositions. However, for that the representation of appositions in the SN, as provided by the parser, could be improved since the order of the two concepts in a sentence is not clear by regarding only the SN, i.e., from the expression *the instrument cello* both SUB0(*instrument.1.1*, *cello.1.1*) and SUB0(*cello.1.1*, *instrument.1.1*) could be extracted. The incorrect relation hypothesis has to be filtered out (hopefully) by the validation component. A better representation would be by employing the TUPL*($c_1, \ldots, c_n$) predicate which combines several concepts with regard to their order. So the example expression should better be represented by SUB(d, e) ∧ TUPL*(e, *instrument.1.1*, *cello.1.1*).

Precision values for the hyponymy relation hypotheses extracted by the learned patterns, which are applied on a subset of the German Wikipedia, are given in Table 2. The first precision value specifies the overall precision, the second the precision if only hypernymy hypotheses are considered which were extracted from first sentences of Wikipedia articles. The precision is usually increased considerably if only such sentences are regarded. Note that this precision value was not given for pattern D_5 which usually cannot be matched to such sentences. The last number specifies the total amount of sentences a pattern could be matched to.

Furthermore, besides the pattern extraction process, the entire hypernymy acquisition system was validated, too. In total 391 153 different hypernymy hypotheses were extracted employing 22 deep and 19 shallow patterns. 149 900 of the relations were only determined by the deep but not by the shallow patterns which shows that the recall can be considerably increased by using deep patterns in addition. But also precision profits from the usage of deep patterns. The average precision of all relations extracted by both shallow and deep patterns is 0.514 that is considerably higher than the average precision for the relations only extracted by shallow patterns (0.243).

The correctness of an extracted relation hypothesis is given for several confidence score intervals in Table 3. There are 89 944 concept pairs with a score above 0.7, 3 558 of them were annotated with the information of whether the hypernymy relation actually holds.

Note that recall is very difficult to specify since for doing this the number of hypernymy relations which are theoretically extractable

from a text corpus has to be known where different annotators can have very dissenting opinions about this number. Thus, we just gave the number of relation hypotheses exceeding a certain score. However the precision obtained by our system is quite competitive to other approaches for hypernymy extraction like the one of Erik Tjong and Kim Sang which extracts hypernyms in Dutch (Tjong and Sang, 2007) (Precision: 0.48).

9 Conclusion and Outlook

We showed a method to automatically derive patterns for hypernymy extraction in form of SNs by following the MDL principle. A list of such patterns together with precision and number of matches were given to show the usefulness of the applied approach. The patterns were applied on the Wikipedia corpus to extract hypernymy hypotheses. These hypotheses were validated using several features. Depending on the score, an arbitrary high precision can be reached. Currently, we determine confidence values for the precision values of the pattern example. Further future work includes the application of our learning algorithm to larger text corpora in order to find additional patterns. Also an investigation of how this method can be used for other types of semantic relations is of interest.

Acknowledgements

We want to thank all of our department which contributed to this work, especially Sven Hartrumpf and Alexander Pilz-Lansley for proofreading this paper. This work was in part funded by the DFG project *Semantische Duplikatserkennung mithilfe von Textual Entailment* (HE 2847/11-1).

References

Cimiano, P. et al. 2005. Learning taxonomic relations from heterogeneous sources of evidence. In Buitelaar, P. et al., editors, *Ontology Learning from Text: Methods, evaluation and applications*, pages 59–73. IOS Press, Amsterdam, The Netherlands.

Cook, D. and L. Holder. 1994. Substructure discovery using minimum description length and background knowledge. *Journal of Artificial Intelligence Research*, 1:231–255.

Fellbaum, C., editor. 1998. *WordNet An Electronic Lexical Database*. MIT Press, Cambridge, Massachusetts.

Hamp, B. and H. Feldweg. 1997. Germanet - a lexical-semantic net for german. In *Proc. of the ACL workshop of Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, Madrid, Spain.

Hartrumpf, S. et al. 2003. The semantically based computer lexicon HaGenLex – Structure and technological environment. *Traitement automatique des langues*, 44(2):81–105.

Hartrumpf, S. 2002. *Hybrid Disambiguation in Natural Language Analysis*. Ph.D. thesis, Fern-Universität in Hagen, Fachbereich Informatik, Hagen, Germany.

Hearst, M. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. of COLING*, Nantes, France.

Helbig, H. 2006. *Knowledge Representation and the Semantics of Natural Language*. Springer, Berlin, Germany.

Joho, H. and M. Sanderson. 2007. Document frequency and term specificity. In *Proc. of RIAO*, Pittsburgh, Pennsylvania.

Morin, E. and C. Jaquemin. 2004. Automatic acquisition and expansion of hypernym links. *Computers and the Humanities*, 38(4):363–396.

Rissanen, J. 1989. *Stochastic Complexity in Statistical Inquiry*. World Scientific Publishing Company, Hackensack, New Jersey.

Snow, R. et al. 2005. Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems 17*, pages 1297–1304. MIT Press, Cambridge, Massachusetts.

Tjong, E. and K. Sang. 2007. Extracting hypernym pairs from the web. In *Proceedings of the 45 Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, Prague, Czech Republic.

vor der Brück, T. 2010. Hypernymy extraction using a semantic network representation. *International Journal of Computational Linguistics and Applications (IJCLA)*, 1(1).

Intrinsic Property-based Taxonomic Relation Extraction from Category Structure

DongHyun Choi and Eun-Kyung Kim and Sang-Ah Shim and Key-Sun Choi
Semantic Web Research Center
KAIST
cdh4696, kekeeo, sashim, kschoi@world.kaist.ac.kr

Abstract

We propose a novel algorithm to extract taxonomic (or *isa/instanceOf*) relations from category structure by classifying each category link. Previous algorithms mainly focus on lexical patterns of category names to classify whether or not a given category link is an *isa/instanceOf*. In contrast, our algorithm extracts intrinsic properties that represent the definition of given category name, and uses those properties to classify each category link. Experimental result shows about 5 to 18 % increase in F-Measure, compared to other existing systems.

1 Introduction

1.1 Problem Description

Taxonomies are a crucial component of many applications, including document clustering (Hotho et al., 2003) and database search (Byron et al., 1997). Due to their importance, many studies have examined methods of extracting taxonomic relations automatically - either from unstructured text (Cimiano et al., 2005; Cimiano(2) et al., 2005), or from structured data such as Wikipedia category structures (Ponzetto and Strube, 2007; Nastase and Strube, 2008; Suchanek et al., 2007). Many researchers have attempted to obtain taxonomic relations from unstructured text to constuct a taxonomy, but in most cases such a system shows poor precision and low recall. Approaches to extracting taxonomic relations from structured data show relatively high performance, but to obtain a taxonomy these require huge amounts of structured data. Recently, as large amounts of structured data such as the infoboxes and category structures of Wikipedia or DBpedia (Auer et al., 2007) have become available, an obstacle to this approach has been removed.

Although a category structure does contain some kind of hierarchical structure, in many cases it cannot be considered as an *isa/instanceOf* hierarchy. For example, the article "Pioneer 11[1]" on Wikipedia is categorized under "Radio frequency propagation", which is related to the "Pioneer 11" but is obviously not a taxonomical parent of "Pioneer 11".

In this paper, we propose a method for extracting taxonomic relations from a given category structure. More precisely, for a category link in the given category structure, the algorithm determines whether the link could be considered an *isa/instanceOf* relation, or if the link simply represents a broader term/narrower term/related term relation. For a given category link <A, B>, in which A is the upper category name and B is the lower category/article name, we attempt to get the definition of B to classify the link. More precisely, we analyze the upper categories of B from the given category structure, to get tokens that represents the definition of B. Once we get the tokens, we compare the tokens with the name of A, to classify the given category link. We call the tokens that represent the definition of B "intrinsic tokens" of B; a more precise definition will be presented in section 3.1.

To show the validity of this approach, the algorithm is applied to Wikipedia's category structure,

[1] Pioneer 11 was the probe for second mission of the Pioneer program (after its sister probe Pioneer 10) to investigate Jupiter and the outer solar system.

Proceedings of the 6th Workshop on Ontologies and Lexical Resources (Ontolex 2010), pages 48–57,
Beijing, August 2010

to obtain taxonomic relations there. Wikipedia's category structure consists of categories, article titles and links between them. A Wikipedia article represents one document, and a category is the grouping of those articles by non-categorization-expert users. Each category has its own name, which is assigned by these users.

Although Wikipedia's category structure is built by non-experts, it can be thought of as reliable since it is refined by many people, and it contains 35,904,116 category links between 764,581 categories and 6,301,594 articles, making it a perfect target for an experimental taxonomic relation extraction algorithm.

After describing related works in section 2, our detailed algorithm is proposed in section 3, and its experimental results are discussed in section 4. In section 5, we make some conclusions and proposals for future work.

2 Related Works

Methods of taxonomic relation extraction can be divided into two broad categories depending on the input: unstructured or structured data. The extraction of taxonomic relations from unstructured text is mainly carried out using lexical patterns on the text. The Hearst pattern (Hearst, 1992) is used in many pattern-based approaches, such as Cimiano (2005).

In addition, there has been research that attempted to use existing structured data, like the Wikipedia category structure or the contents of a thesaurus. The system of Ponzetto (2007) determines whether or not the given Wikipedia category link is an *isa/instanceOf* relation by applying a set of rules to the category names, while Nastase (2008) defined lexical patterns on category names, in addition to Ponzetto (2007). The YAGO system (Suchanek et al., 2007) attempts to classify whether the given article-category link represents an *instanceOf* relation by checking the plurality of the upper category name.

The algorithm proposed in this paper focuses on the structured data, mainly the category structure, to gather *isa/instanceOf* relations. The system gets a category structure as input, and classifies each category link inside the category structure according to whether it represents an *isa/instanceOf* relation or not.

3 Algorithm Description

In section 3.1, we introduce the necessary definitions for *isa/instanceOf* relations and the required terms to describe the algorithm. In section 3.2, we will discuss the hypotheses based on the definitions described in section 3.1. Next, two binary classification algorithms will be proposed based on the hypotheses, which will determine whether the given category link is an *isa/instanceOf* relation or not.

3.1 Definitions

To define *isa* and *instanceOf* relations, Mizoguchi (2004) introduces the concept of *intrinsic property* and other related concepts, which are shown in the following definitions 1, 2 and 3:

Definition 1: **Intrinsic property.** The intrinsic property of a thing is a property which is essential to the thing and it loses its identity when the property changes.

Definition 2: **The ontological definition of a class.** A thing which is a conceptualization of a set X can be a class if and only if each element x of X belongs to the class X if and only if the intrinsic property of x satisfies the intensional condition of X. And, then and only then, <x *instanceOf* X> holds.

Definition 3: isa **relation.** *isa* relation holds only between classes. <class A *isa* class B> holds iff the instance set of A is a subset of the instance set of B.

In addition, we define the following terms for algorithm description:

Definition 4: **intrinsic token.** Token [2] T is an intrinsic token of B iff T represents the intrinsic property of B.

For example, when B is "Pioneer 11", the intrinsic tokens of B are "spacecraft", "escape[3]", "Jupiter", etc.

[2]For example, token is a segmented term in category names of Wikipedia category structure.

[3]Since the main purpose of Pioneer 11 is to escape from the solar system and fly into the deep space, we thought "escape" is the intrinsic token of "Pioneer 11". In the same context, "spacecraft escaping the solar system" is a taxonomical parent of "Pioneer 11".

Definition 5: **category link.** <A, B> is called category link iff A is a category of B, and that fact is explicitly stated in the given category structure.

Consider the example of Wikipedia. If B is an article, <A, B> is called an **article-category link**, and if B is a category, <A, B> is called a **category-category link**. The article is a categorized terminal object.

Definition 6: **category structure.** Category structure is the collection of category links, its component categories, and categorized terminal objects.

Definition 7: **upper category set.** The upper category set of B is defined as the set of upper categories of B up to n step in the given category structure, and it is expressed as U(B, n).

For example, if the two category links <*Jupiter spacecraft, Pioneer 11*> and <*Jupiter, Jupiter spacecraft*> exist inside the given category structure, then *Jupiter spacecraft* is the element of U(Pioneer 11, 1), while *Jupiter* is not.

Figure 1 shows the category structure of U(Pioneer 11, 3),which we refer to throughout this paper to explain our algorithm.

3.2 Hypotheses

According to the classical Aristotelian view, categories are discrete entities characterized by a set of properties shared by their members. Thus, we make the following lemmas:

Lemma 1: If some objects are grouped into the same category, then they share at least more than one property.

According to definition 2, if x is an *instanceOf* X, then the intrinsic property of x satisfies the definition of X. Since the intrinsic property is the property related to the definition of the object, we can assume that in most categorization systems, the intrinsic property is the most frequently shared property among those objects categorized in the same category.

Lemma 2. Intrinsic properties are shared most frequently among objects in a category.

Lemma 2 means that, for example, the intrinsic token T of B will show up frequently among the names of upper categories of B. But lemma 2 does NOT mean that non-intrinsic tokens will not frequently appear among the upper category names. For example, the elements of U(Pioneer 11, 3) from the Wikipedia category structure contain the token "spacecraft" 4 times, but it also contain token "technology" 3 times. Therefore, we cannot directly use the token frequency to determine which one is the intrinsic token: rather, we make another assumption to get the "intrinsic score" for each token.

Lemma 3. Intrinsic tokens co-occur frequently with other intrinsic tokens.

Lemma 3 means that, if T1 is an intrinsic token of B, and T2 co-occurs with T1 inside the upper category names of B, then there is a high probability that T2 is also an intrinsic token of B. For example, for the category link <Jupiter spacecraft, Pioneer 11>, if the token "spacecraft" is an intrinsic token of "Pioneer 11", we can assume that the token "Jupiter" is also an intrinsic token of "Pioneer 11". Since some intrinsic tokens that are appropriate as modifiers are not appropriate as head words – for example, if the token "Jupiter" is used as a modifier, it will be a *good* intrinsic token of "Pioneer 11", but if it is used as a head word, choosing it as the intrinsic token of "Pioneer 11" would be *bad* choice – thus, we distinguish between intrinsic score as head word, and intrinsic score as modifier. If the intrinsic score of token T is high for article/category name B, then it means the probability is high that T is an intrinsic token of B. We assumed that only the co-occurrences as head word and its modifier are meaningful. **Corollary 3-1.** If a modifier co-occurs with a head word, and the head word is frequently an intrinsic token of an object, then the modifier is an intrinsic token of the object.

Corollary 3-2. If a head word co-occurs with a modifier, and the modifier is frequently an intrinsic token of an object, then the head word is an intrinsic token of the object.

3.3 Proposed Algorithm

Based on the hypotheses proposed in section 3.2, we propose two algorithms to get the intrinsic score of each token in the following sections. The first algorithm, a counting-based approach, uses only lemmas 1 and 2, and it will be shown why this algorithm will not work. The second algorithm, a graph-based approach, uses all of the hy-

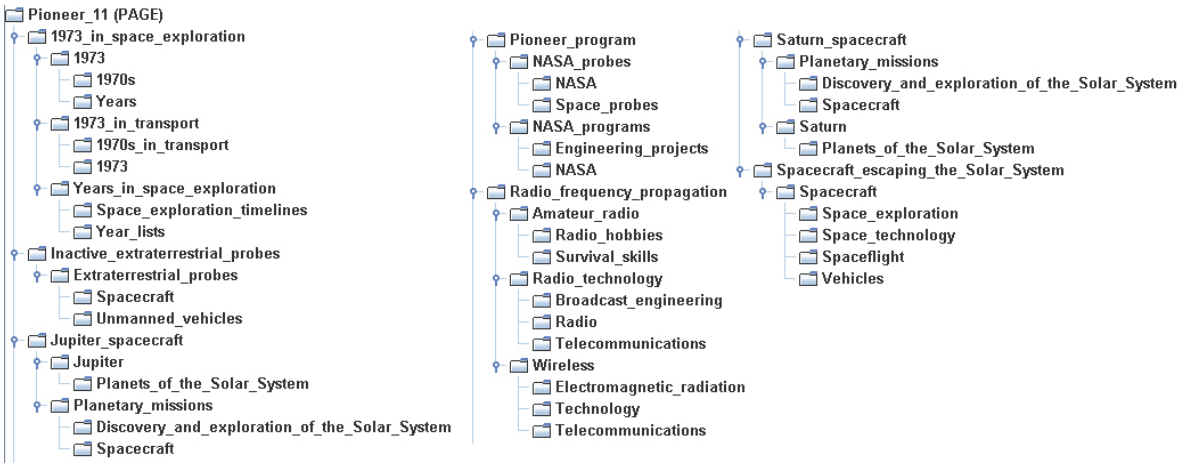

Figure 1: category structure of U(Pioneer 11, 3) from Wikipedia.

potheses to solve the problem.

For the given category link <A, B>, the intrinsic score of each token will be calculated based on its frequency inside U(B, n) while separately counting the token's intrinsic score as modifiers and the intrinsic score as head word. We here propose a scoring mechanism based on the HITS page ranking algorithm (Kleinberg, 1999): For the given category link <A, B>, we first construct a "modifier graph" using U(B, n), and then calculate the intrinsic score for each token in U(B, n) using the HITS algorithm. After that, the intrinsic score of each token will be used to calculate the score of <A, B>. If the score is higher than some predefined threshold, then <A, B> is classified as an *isa/instanceOf* link, and otherwise it is not.

3.3.1 Counting-based Approach

This method utilizes lemmas 1 and 2 to get the intrinsic score for each token, and then uses the score to determine whether the given category link is an *isa/instanceOf* link or not.

To utilize this approach, we first score each token from U(B, n) by counting the frequency of each token from the words of U(B, n). Table 1 shows the score of each token from U(Pioneer, 3) for figure 1.

For the "Pioneer 11" article, there are seven category links in Wikipedia's category structure: <1973 in space exploration, Pioneer 11>, <Inactive extraterrestrial probes, Pioneer 11>, <Jupiter spacecraft, Pioneer 11>, <Pioneer program, Pioneer 11>, <Radio frequency propagation, Pioneer 11>, <Saturn spacecraft, Pioneer 11>, and <Spacecraft escaping the Solar System, Pioneer 11>, as shown in figure 1. The scores of each link using a counting-based approach are acquired by adding the scores for each token in table 1 that is matched with single term occurrence in category names. Table 2 shows the result of counting-based approach.

Token	**Score**
space	6
exploration	5
spacecraft, probe	4
1973, technology, year, radio, solar, system, nasa	3
vehicle, radio, program, 1970s, extraterrestrial, transport, Saturn, Jupiter	2
escape, inactive, frequency, propagation, pioneer, ...	1

Table 1: Score for each token from U(Pioneer 11, 3)

Although the link <1973 in space exploration, Pioneer 11> receives the highest score among those seven links, obviously the link does not represent *isa/instanceOf* relation. This shows that the counting approach does not guarantee accuracy. Table 1 shows that non-intrinsic tokens occur frequently (such as 'technology' in this exam-

Article-Category Links	Score
<1973 in space exploration, Pioneer 11>	3+6+5=14
<Spacecraft escaping the Solar System, Pioneer 11>	4+1+3+3=11
<Inactive extraterrestrial probes, Pioneer 11>,	1+2+4=7
<Saturn spacecraft, Pioneer 11>	2+4=6
<Jupiter spacecraft, Pioneer 11>	2+4=6
<Radio frequency propagation, Pioneer 11>	2+1+1=4
<Pioneer program, Pioneer 11>	1+2=3

Table 2: Scoring each category links using counting approach

ple). We call this an 'overloaded existence' error. To solve the problems described above, we apply Lemma 3, Corollary 3-1 and 3-2 to our calculation, and propose a second algorithm based on a graph-based approach, which will be explained in the next section.

3.3.2 Graph-based Approach

In this section, we propose a graph-based approach to get the intrinsic score of each token. To do this, we first construct a modifier graph from the words of U(B, n) for a given category link <A, B>, with each node representing a token from the elements of U(B, n), and each edge representing the co-occurrence of tokens inside each element of U(B, n). Next, we apply a well-known graph analysis algorithm to that graph, and get the intrinsic scores for each node. Finally, we use the score of each node to get the score of the given category link.

Constructing modifier graph Modifier graph constructed here is defined as a directed graph, in which each node represents each token inside U(B, n), and each edge represents a co-occurrence as modifier-head relation inside each category name of U(B, n). Using the subset of U(Pioneer 11, 3), we get the modifier graph of figure 2.[4]

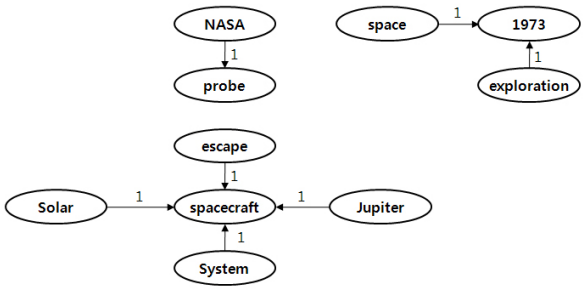

Figure 2: Modifier graph of the subset of U(Pioneer 11, 3): {Spacecraft escaping the Solar System, Jupiter spacecraft, 1973 in space exploration, NASA probes, Saturn}

Calculating Intrinsic score After constructing the modifier graph, we apply the HITS algorithm to the modifier graph. Since the HITS algorithm cannot reflect the weight of edges, a modified version of the HITS algorithm (Mihalcea and Tarau, 2005) is adopted:

$$\textit{Authority}(V_i) = \sum_{V_j \in In(V_i)} e_{ji} \cdot \textit{Hub}(V_j) \quad (1)$$

$$\textit{Hub}(V_i) = \sum_{V_j \in Out(V_i)} e_{ij} \cdot \textit{Authority}(V_j) \quad (2)$$

$In(V_i)$ represents the set of vertices which has the outgoing edge to V_i, $Out(V_i)$ represents the set of vertices which has the incoming edge from V_i, and e_{ij} represents the weight of the edge from V_i to V_j. The algorithm for calculating the scores is as follows:

1. Initialize the authority and hub score of each node to one.

2. Calculate hub score of each node using the formula 2.

3. Calculate authority score of each node using the formula 1.

4. Normalize authority & hub score so that the sum of authority score of every node and the sum of hub score of every node are one.

[4]We used the full set of U(B, n) to create the modifier graph for the full scale of experimentation in section 4.

5. Iterate from step 2 until the score of every node converges.

In the modifier graph, Authority score can be mapped to the intrinsic score of a node(token) as a head word, and Hub score can be mapped to the intrinsic score of a node(token) as a modifier.

Scoring Category Link Now, we can score the input category link. The score of category link <A, B> is given as follows:

$$Score(< A, B >) = Authority(h) + \sum_{a \text{ in } mod(A)} Hub(a) \quad (3)$$

Here, $Score(< A, B >)$ represents the final score of category link <A, B>, h represents the head word of A, and $mod(A)$ represents the set of modifiers of A. Since the score of head word and modifiers are calculated based on the upper categories of B, this formula can integrate both meaning of A and B to classify whether the link is isa/instanceOf. Table 3 shows the scores of seven article-category links from table 2, calculated using the graph-based approach.

Article-Category Links	Score
<Spacecraft escaping the Solar System, Pioneer 11>	0.5972
<1973 in space exploration, Pioneer 11>	0.4018
<Jupiter spacecraft, Pioneer 11>	0.2105
<Saturn spacecraft, Pioneer 11>	0.2105
<Inactive extraterrestrial probes, Pioneer 11>,	0.0440
<Radio frequency propagation, Pioneer 11>	0.0440
<Pioneer program, Pioneer 11>	0.0132

Table 3: Scoring each category links using graph-based approach

The link <Spacecraft escaping the Solar System, Pioneer 11> gets the highest score, while the link <1973 in space exploration, Pioneer 11>, which got the highest score using counting-based approach, gets the second place. That proves the algorithm's effectiveness for distinguishing *isa/instanceOf* link from other non-*isa/instanceOf* links. But there is still a problem - although the first-ranked link is a *isa/instanceOf* link, the second-ranked is not, while the third and fourth-ranked links (<Jupiter spacecraft, Pioneer 11>, <Saturn spacecraft, Pioneer 11> are *isa/instanceOf* links. To get a better result, we propose four additional modifications in the next secton.

3.4 Additional Modifications to the Graph-based Approach

To better reflect the category structure and the property of category names to the scoring mechanism, the following four modifications can be made. Each of these modification could be applied independently to the original algorithm described in section 3.3.2.

Authority Impact Factor (I). In most cases, a category name contains only one head word, while it contains 2 or more modifiers. As Formula (3) is just the linear sum of the hub scores of each modifier and the authority score of the head word, the resultant score is more affected by hub score, because the number of modifiers is normally bigger than the number of head words. To balance the effect of hub score and authority score, we introduce authority impact factor *I*:

$$Score(< A, B >) = I \cdot Authority(h) + \sum_{a \text{ in } mod(A)} Hub(a) \quad (4)$$

The authority impact factor is defined as the average number of modifiers in the elements of U(B, n), since normally each category name contains only one head word.

Dummy Node (D). There are some category names that contain only one head word and no modifier, thus making it impossible to create the modifier graph.[5] Thus, for such category names we introduce dummy nodes to include their information into the modifier graph. In figure 3, you can observe the introduction of the dummy node 'dummy0'.

[5]For example, in figure 2, we cannot find node 'Saturn' while U(Pioneer 11, 3) contains category name 'Saturn'

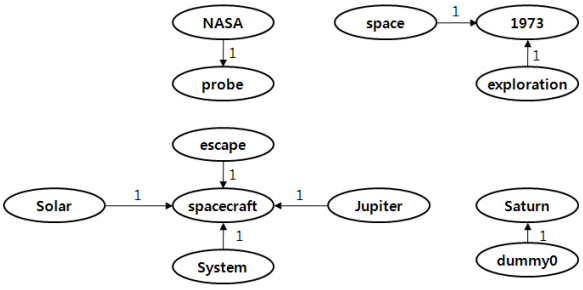

Figure 3: Modifier graph of the subset of U(Pioneer 11, 3), with dummy node.

Category Distance Factor (C). We define the category distance between category/article A and B as the minimum number of category links required to reach B from A by following the category links. Category distance factor *C* of a category name A from U(B, n) is the reverse of the category distance between A and B. We assumed that, if the distance between A and B is higher, then it is less probable for A to have the intrinsic property of B. Based on this assumption, category distance factor *C* of category name A is multiplied by the edge score of an edge generated by category name A.

Figure 4 shows the modifier graph of figure 2 that applies the category distance factor. Since the category distance between "Pioneer 11" and "NASA probe" is two, the score of edge (NASA, probe) is 1/2 = 0.5.

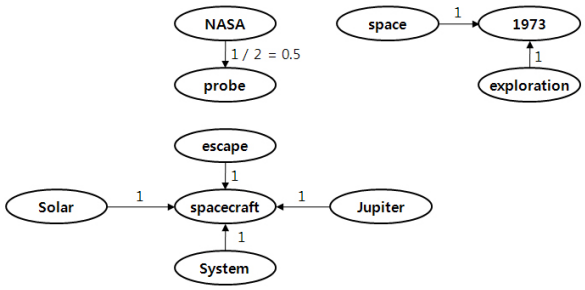

Figure 4: Modifier graph of the subset of U(Pioneer 11, 3), with category distance factor.

Modifier Number Normalization Factor (W). In the algorithm of building a modifier graph, the head word of a category name with many modifiers has the advantage over the head word of a category name with few modifiers, as if a category name contains n modifiers it will generate n edges incoming to its head word. To overcome this problem, we defined the modifier number normalization factor *W* for each category name: it is defined as the reverse of the number of modifiers in the category name, and it is multiplied by the edge score of an edge, generated by the category name, of the modifier graph. Figure 5 shows the modifier graph of figure 2 with the modifier number normalization factor. Since the category name "Spacecraft escaping the Solar System" has three modifiers, the scores of edge (escape, Pioneer 11), (solar, Pioneer 11) and (system, Pioneer 11) are 1/3 = 0.33.

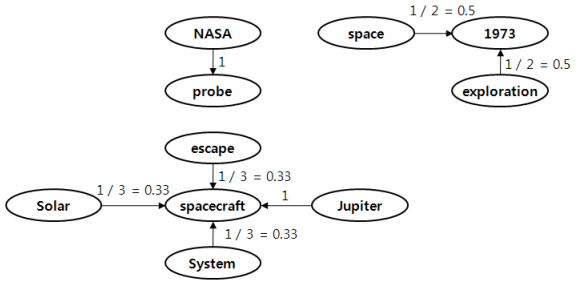

Figure 5: Modifier graph of the subset of U(Pioneer 11, 3), with modifier number normalization factor.

Removing roleOf Relation (E). To distinguish the roleOf relation from taxonomic relation,we introduce a new *E*. This feature simply classify the link <A, B> as non-*instanceOf* if category name A has endings like -er, -ers, -or, -ors, -ian, -ians. Since only the terminal node can represent the name of person in category structure, we applied this feature to classify only article-category links. One of the example from Wikipedia which should be judged as roleOf relation is <La Liga footballer, Cristiano Ronaldo>.

After applying above four modifications, we get the result in table 4. Now, top 3 links all represent *instanceOf* links.

Article-Category Links	Score
<Spacecraft escaping the Solar System, Pioneer 11>	2.1416
<Jupiter spacecraft, Pioneer 11>	2.1286
<Saturn spacecraft, Pioneer 11>	2.1286
<1973 in space exploration, Pioneer 11>	0.0241
<Pioneer program, Pioneer 11>	0.0062
<Inactive extraterrestrial probes, Pioneer 11>,	0.0026
<Radio frequency propagation, Pioneer 11>	0.0021

Table 4: Scoring each category links using graph-based approach with four modifications.

4 Implementation

We implemented a combinatory system that combines the algorithm suggested by this paper with existing lexical pattern-based algorithms. More precisely, we set two parameters α and β, in which β has a consistently higher value than α. If score of the given category link, which is retrieved by the proposed system, is higher than β, it is classified as *isa/instanceOf*. If the score is higher than α but lower or equal to β, the system uses an existing lexical pattern-based algorithm to classify the link. If the score is lower than or equal to α, it is classified as not isa/instanceOf.

To test the system, we used Wikipedia's category structure, which contains 1,160,248 category-category links and 15,778,801 article-category links between 505,277 categories and 6,808,543 articles. We extract category links from the Wikipedia category structure and annotate them to construct the test corpus. During the process of choosing category links, we intentionally removed category links with names containing any of the following words: "stub", "wikiproject", "wikipedia", "template", "article", "start-class", "category", "redirect", "mediawiki", "user", "portal", "page", and "list". These words are normally used to represent Wikipedia maintenance pages. After we remove the links described before, we randomly choose 3,951 category-category links and 1,688 article-category links. Two annotators worked separately to annotate whether or not the given link is an *isa/instanceOf* link, and in the event of conflict they would discuss the case and make a final decision.

We carried out experiments on category-category link set and article-category link set separately, since their characteristics are different. We assumed that the taxonomic relation in a category-category link is an *isa* link, while the taxonomic relation in an article-category link is an *instanceOf* link. To acquire the upper category set, we set n=3 throughout the experiment. For head word extraction, the method of Collins (1999) is used, and for lemmatization we used the Lingpipe toolkit (Alias-i, 2008).

4.1 Experiments on category-category link

We divided the 3,951 category-category links into two equally-sized sets, and used one set as a training set and the other one as a test set. The training set was used to identify the α and β values for *isa* link classification: in other words, the α and β values that showed the best performance when applied to training set were selected as the actual parameters used by the system. As Wikipedia's category structure contains a huge number of category links, precision is more important than recall. As recall cannot be ignored, we chose the parameters that gave the highest precision on the training set, while giving a recall of at least 0.7. Also, we carried out experiments on three baseline systems.The first one determined every link as an *isa* link. The second one applied the head word matching rule (M) only, which says that for category-category link <A, B>, if the head words of A and B are the same, then <A, B> should be classified as an *isa* link. The third one applies the method of Ponzetto (P) (Ponzetto and Strube, 2007). The ruleset of Ponzetto includes Head word matching rule, Modifier-head word matching rule(Ex. <Crime, Crime Comics>: Head word of "Crime" and modifier of "Crime Comics" matches: Not *isa*), and the plurality rule used by YAGO system(Explained at the next chapter)).

Table 5 shows the baseline results, the results of existing systems, and our best results on the test set. Usage of authority score is represented as *A*, and usage of hub score is represented as *H*. Also,we did experiments on all possible combina-

tion of features A, H, I, D, C, W, M, P. For example, Comb(AHICDM) means that we used feature A, H, I, C, D to construct the modifier graph and score the category link, and for those whose score is between α and β we used head word matching rule to classify them. At the table, P stands for Precision, R stands for Recall, and F stands for F-measure.

Setting	P	R	F
Baseline1	0.7277	1.0	0.8424
Baseline2(M)	0.9480	0.6335	0.7595
Baseline3(P)	0.9232	0.6516	0.7640
Comb1(AHM)	0.9223	0.7350	0.8181
Comb2(AHP)	0.8606	0.7211	0.7847
Comb3(AHICM)	0.9325	0.7302	0.8190

Table 5: Experimental result on test set of category-category links: Baseline vs. System best result

As you can observe, the precision of head-word matching (M) is high, meaning that in many cases the head word represents the intrinsic property. Also, its recall shows that for category-category links, at least more than half of the categories are categorized using the intrinsic property of the objects grouped within them, which strongly supports lemma 2 in section 3.2. The comparison of setting M and AHM, P and AHP shows that the intrinsic-property based approach increases recall of the existing system about 7-10 %, at the cost of of 2-6 % precision loss. This shows that, rather than looking only at the given category link and analyzing patterns on its name, by gathering information from the upper category set, we were able to significantly increase recall. However, it also shows that some "garbage" information is introduced through the upper category set, resulting in a 2-6 % precision loss. The best system shows about a 8-10 % increase in recall, with comparably good precision compared to the two baseline systems.

4.2 Experiments on article-category link

In a similar manner to the experiments on category-category links, we divided the 1,688 article-category links into two equally-sized sets, and used one set as a training set and the other one as a test set. The training set is used to determine the parameters for *instanceOf* link classification. The parameter setting procedure was the same as in the experiments on category-category links, except that we used the article-category links for the procedure. In this experiment, we also adapted three baseline systems. The first system classifies every link as an *instanceOf* link, the second system adapts the head word matching rule (M), and the third system applies the rule from Yago (Y) (Suchanek et al., 2007), which states that for article-category link <A, B>, if A is plural then the link could be classified as an *instanceOf* relation.

Setting	P	R	F
Baseline1	0.5261	1.0	0.6894
Baseline2(M)	0.7451	0.0856	0.1535
Baseline3(Y)	0.6036	0.5315	0.5653
Comb1(AHY)	0.6082	0.6718	0.6381
Comb2(ADWEY)	0.7581	0.7410	0.7494

Table 6: Experimental result on test set of article-category links on some settings

Table 6 shows the baseline results and the best results of the combinatory system. As you can observe from the above table, M (head word matching rule) does not work well in article-category links, although its precision is still high or comparable to that of other methods. Since in most cases an article represents one instance, in many cases they have their own name, making the recall of the head word matching rule extremely low. Also, the combination system 1 (AHY) shows comparable precision with Y but 14 % higher in reall, resulting 7 % increse in F-Measure.The best system shows about 18 % increase in F-measure, especially 15 % precision increase and 21 % recall increase compared to YAGO system.

5 Conclusion and Future work

In this paper, we explored a intrinsic token-based approach to the problem of classifying whether a category link is a taxonomic relation or not. Unlike previous works that classify category links, we acquired the definition of a lower category

name by extracting intrinsic tokens and using them to score the given category link, rather than by applying predefined lexical rules to the category link. Our intrinsic token-based approach leads to a significant improvement in F-measure compared to previous state-of-the-art systems. One possible future direction for research is automatic instance population, by using those extracted intrinsic tokens and gathering taxonomic relations from the category structure.

Acknowledgments

This work was supported by the Industrial Strategic Technology Development Program (10035348, Development of a Cognitive Planning and Learning Model for Mobile Platforms) funded by the Ministry of Knowledge Economy(MKE, Korea).

References

Soumen C. Byron, Byron Dom, Rakesh Agrawal, and Prabhakar Raghavan. 1997. Using taxonomy, discriminants, and signatures for navigating in text databases. *Proceedings of the international conference on very large data bases*, 446–455.

Philipp Cimiano, Andreas Hotho, and Steffen Staab. 2005. Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. *Journal of Artificial Intelligence Research*, 24:305–339.

Philipp Cimiano, Aleksander Pivk, Lars Schmidt-Thieme, and Steffen Staab. 2005. Learning Taxonomic Relations from Heterogeneous Sources of Evidence. *Ontology Learning from Text: Methods, Evaluation and Applications*, 59–73.

Marti A. Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. *Proceedings of the 14th conference on Computational linguistics*, 2:539–545.

Andreas Hotho, Steffen Staab, and Gerd Stumme. 2003. Ontologies improve text document clustering. *Proceedings of the IEEE International Conference on Data Mining*, 541–544.

Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632

Simone P. Ponzetto, and Michael Strube. 2007. Deriving a Large Scale Taxonomy from Wikipedia. *Proceedings of the AAAI07*.

Vivi Nastase, and Michael Strube. 2008. Decoding Wikipedia category names for knowledge acquisition . *Proceedings of the AAAI08*.

Riichiro Mizoguchi. 2004. Part 3: Advanced course of ontological engineering. *New Generation Computing*, 22(2): 193–220

Rada Mihalcea, and Paul Tarau. 2005. A Language Independent Algorithm for Single and Multiple Document Summarization. *Proceedings of IJCNLP 2005*.

Ian Niles, and Adam Pease. 2003. Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology. *Proceedings of the IEEE International Conference on Information and Knowledge Engineering*.

Soeren Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. *Lecture Notes in Computer Science*, 4825/2007:722–735.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia *Proceedings of the 16th international conference on World Wide Web*, 697–706.

Michael Collins. 1999. Head-driven Statistical Models for Natural Language Parsing. *University of Pennsylvania PhD Thesis*.

Alias-i. 2008. LingPipe 3.9.1. *http://alias-i.com/lingpipe*.

Developing a Biosurveillance Application Ontology for Influenza-Like-Illness

Mike Conway, John Dowling and Wendy Chapman
Department of Biomedical Informatics
University of Pittsburgh
{conwaym|dowling|wec6}@pitt.edu

Abstract

Increasing biosurveillance capacity is a public health priority in both the developed and the developing world. Effective *syndromic* surveillance is especially important if we are to successfully identify and monitor disease outbreaks in their early stages. This paper describes the construction and preliminary evaluation of a syndromic surveillance orientated application ontology designed to facilitate the early identification of Influenza-Like-Illness syndrome from Emergency Room clinical reports using natural language processing.

1 Introduction and Motivation

Increasing biosurveillance capacity is a public health priority in both the developed and developing world, both for the early identification of emerging diseases and for pinpointing epidemic outbreaks (Chen et al., 2010). The 2009 Mexican flu outbreak provides an example of how an outbreak of a new disease (in this case a new variant of H1N1 influenza) can spend some weeks spreading in a community before it is recognized as a threat by public health officials.

Syndromic surveillance is vital if we are to detect outbreaks at an early stage (Henning, 2004; Wagner et al., 2006). The United States Center for Disease Control (CDC) defines syndromic surveillance as "surveillance using health-related data that precede diagnosis and signal a sufficient probability of a case or outbreak to warrant further public health response."[1] That is, the focus of syndromic surveillance is the identification of disease outbreaks *before* the traditional public health apparatus of confirmatory laboratory testing and official diagnosis can be used. Data sources for syndromic surveillance have included, over the counter pharmacy sales (Tsui et al., 2003), school absenteeism records (Lombardo et al., 2003), calls to *NHS Direct* (a nurse led information and advice service in the United Kingdom) (Cooper, 2007), and search engine queries (Eysenbach, 2006).

However, in this paper we concentrate on mining text based clinical records for outbreak data. Clinical interactions between health workers and patients generate large amounts of textual data — in the form of clinical reports, chief complaints, and so on — which provide an obvious source of pre-diagnosis information. In order to mine the information in these clinical reports we are faced with two distinct problems:

1. How should we define a syndrome of interest? That is, how are signs and symptoms mapped to syndromes?

2. Given that we have established such a set of mappings, how then do we map from the text in our clinical reports to the signs and symptoms that constitute a syndrome, given the high level of terminological variability in clinical reports.

This paper presents an application ontology that attempts to address both these issues for the domain of Influenza-Like-Illness Syndrome (ILI). The case definition for ILI, as defined by the United States Center for Disease Control is "fever greater than or equal to 100 degrees Fahrenheit

[1] www.webcitation.org/5pxhlyaxX

Proceedings of the 6th Workshop on Ontologies and Lexical Resources (Ontolex 2010), pages 58–66,
Beijing, August 2010

and either cough or sore throat."[2] In contrast to the CDC's straightforward definition, the syndrome is variously described as a cluster of symptoms and findings, including fever and cold symptoms, cough, nausea, vomiting, body aches and sore throat (Scholer, 2004). In constructing an application specific syndrome definition for this ontology, we used a data driven approach to defining ILI, generating a list of terms through an analysis of Emergency Room reports.

The remainder of the paper is divided into five parts. First, we briefly describe related work, before going on to report on the ontology development process. We then set forth an evaluation of the ontology with respect to its coverage of terms in the target domain. We go on to outline areas for future work, before finally presenting some concluding comments.

2 Related Work

In recent years there has been significant progress in interfacing lexical resources (in particular WordNet (Miller, 1995)) and upper level ontologies (like the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) (Gangemi et al., 2002) and the Suggested Upper Merged Ontology (SUMO) (Niles and Pease, 2003)). However, as our domain of interest employs a highly specialized terminology, the use of general linguistic resources like WordNet was inappropriate.

Our work has focused on the representation of ILI relevant concepts that occur in clinical reports in order to facilitate syndromic surveillance. While the widely used medical taxonomies and nomenclatures (for example Unified Medical Language System[3] and the Systematized Nomenclature of Medicine Clinical Terms[4]) contain many of the ILI relevant concepts found in clinical texts, these general resources do not have the specific relations (and lexical information) relevant to syndromic surveillance from clinical reports. Currently, there are at least four major terminological resources available that focus on the public health domain: PHSkb, SSO, and the BioCaster Ontology.

[2] www.webcitation.org/5q22KTcHx

[3] www.nlm.nih.gov/research/umls/

[4] www.ihtsdo.org/snomed-ct/

2.1 PHSkb

The Public Health Surveillance knowledge base PHSkb (Doyle et al., 2005) developed by the CDC is a coding system for the communication of notifiable disease[5] findings for public health professionals at the state and federal level in the United States. There are however several difficulties in using the PHSkb directly in an NLP orientated syndromic surveillance context:

1. Syndromic surveillance requires that syndromes and signs are adequately represented. The PHSkb emphasizes *diagnosed* diseases. That is, the PHSKb is focused on post diagnosis reporting, when laboratory tests have been conducted and the presence of a disease is confirmed. This approach is not suitable for syndromic surveillance where we seek to identify clusters of symptoms and signs *before* a diagnosis.

2. PHSkb is no longer under active development.

2.2 SSO

The Syndromic Surveillance Ontology (SSO) (Okhmatovskaia et al., 2009) was developed to address a pressing problem for system developers and public health officials. How can we integrate outbreak information when every site uses different syndrome definitions? For instance, if State **X** defines *sore throat* as part of ILI, yet State **Y** does not, syndromic surveillance results from each state will not be directly comparable. When we apply this example to the wider national scene, with federal regional and provincial public health agencies attempting to share data with each other, and international agencies, we can see the scale of the problem to be addressed.

In order to manage this data sharing problem, a working group of eighteen researchers, representing ten functional syndromic surveillance systems in the United States (for example, Boston Public Health Department and the US Department of Defense) convened to develop standard

[5] A notifiable disease is a disease (or by extension, condition) that must, by law, be reported to the authorities for monitoring purposes. In the United States, examples of notifiable diseases are: Shigellosis, Anthrax and HIV infection.

definitions for four syndromes of interest (*respiratory*, *gastro-intestinal*, *constitutional* and *ILI*)[6] and constructed an OWL ontology based on these definitions. While the SSO is a useful starting points, there are several reasons why — on its own — it is insufficient for clinical report processing:

1. SSO is centered on *chief complaints*. Chief complaints (or "presenting complaints") are phrases that briefly describe a patient's presenting condition on first contact with a medical facility. They usually describe symptoms, refrain from diagnostic speculation and employ frequent abbreviations and misspellings (for example "vom + naus" for "vomiting and nausea"). Clinical texts — the focus of attention in this paper — are full length documents, normally using correct spellings (even if they are somewhat "telegraphic" in style). Furthermore, clinical reports frequently list physical findings (that is, physical signs elicited by the physician, like, for instance reflex tests) which are not present in symptom orientated chief complaints.

2. The range of syndromes represented in SSO is limited to four. Although we are starting out with ILI, we have plans (and data) to extend our resource to four new syndromes (see Section 5 for details of further work).

3. The most distinctive feature of the SSO is that the knowledge engineering process was conducted in a face-to-face committee context. Currently, there is no process in place to extend the SSO to new syndromes, symptoms or domains.

2.3 BioCaster Ontology

The BioCaster application ontology was built to facilitate text mining of news articles for disease outbreaks in several different Pacific Rim languages (including English, Japanese, Thai and Vietnamese) (Collier et al., 2006). However, the ontology, as it stands, is not suitable for supporting text mining clinical reports, for the following reasons:

1. The BioCaster ontology concentrates on the types of concepts found in published news outlets for a general (that is, non medical) readership. The level of conceptual granularity and degree of terminological sophistication is not always directly applicable to that found in documents produced by health professionals.

2. The BioCaster ontology, while it does represent syndromes (for example, constitutional and hemorrhagic syndromes) and symptoms, does not represent physical findings, as these are beyond its scope.

In addition to the application ontologies described above, the Infectious Disease Ontology provides an Influenza component (and indeed wide coverage of many diseases relevant to syndromic surveillance). In Section 5 we describe plans to link to other ontologies.

3 Constructing the Ontology

Work began with the identification of ILI terms from clinical reports by author JD (a board-certified infectious disease physician with thirty years experience of clinical practice) supported by an informatician [author MC]. The term identification process involved the project's domain expert reading multiple reports,[7] searching through appropriate textbooks, and utilizing professional knowledge. After a provisional list of ILI concepts had been identified, we compared our list to the list of ILI concepts generated by the SSO ILI component (see Section 2.2) and attempted to reuse SSO concepts where possible. The resulting ILI concept list consisted of 40 clinical concepts taken from SSO and 15 new concepts. Clinical concepts were divided into three classes: Disease (15 concepts), Finding (21 concepts) and Symptom (19 concepts). Figure 1 shows the clinical

[6] A demonstration chief complaint classifier based on SSO is available at:
`http://onto-classifier.dbmi.pitt.edu/onto_classify.html`

[7] De-identified (that is, anonymized) clinical reports were obtained through partnership with the University of Pittsburgh Medical Center.

concepts covered. As part of our knowledge engineering effort, we identified concepts and associated relations for several different syndromes which we plan to add to our ontology at a later date.[8]

Early on in the project development process, we took the decision to design our ontology in such a way as to maintain consistency with the BioCaster ontology. We adopted the BioCaster ontology as a model for three reasons:

1. A considerable knowledge engineering effort has been invested in BioCaster since 2006, and both the domain (biosurveillance) and application area (text mining) are congruent to our own.

2. The BioCaster ontology has proven utility in its domain (biosurveillance from news texts) for driving NLP systems.

3. We plan to import BioCaster terms and relations, and thus settled on a structure that facilitated this goal.

The BioCaster ontology (inspired by the structure of EuroWordNet[9]) uses *root terms* as interlingual pivots for the multiple languages represented in the ontology.[10] One consequence of following this structure is that all clinical concepts are *instances*.[11] Additionally, all specified relations are relations between instances.

Relations relevant to the syndromic surveillance domain generally were identified by our physician in conjunction with an informatician (MC). Although some of these relations (like `is_bioterrorismDisease`) are less relevant to ILI syndrome, they were retained in order to maintain consistency with planned future work. Additionally, we have added links to other terminological resources (for example, UMLS and Snomed-CT)

[8]Note that finer granularity was used in the initial knowledge acquisition efforts (for example, we distinguished *sign* from *physical finding*).

[9]`http://www.illc.uva.nl/EuroWordNet/`

[10]Note that we are using *root term* instead of the equivalent EuroWordNet term *Inter Lingual Index*.

[11]Note that from a formal ontology perspective, concepts are instantiated in text. For example, "Patient **X** presents with *nausea* and *high fever*" instantiates the concepts **nausea** and **high fever**.

Lexical resources and regular expressions are a vital component of our project, as the ontology has been built with the public health audience in mind (in practice, state or city public health IT personnel). These users have typically had limited exposure to NLP pipelines, named entity recognizers, and so on. They require an (almost) "off the shelf" product that can easily be plugged into existing systems for text analysis.

The ontology currently includes 484 English keywords and 453 English regular expression. The core classes and relations were developed in Protege-OWL, and the populated ontology is generated from data stored in a spreadsheet (using a Perl script). Version control was managed using Subversion, and the ontology is available from a public access Google code site.[12] Figure 2 provides a simplified example of relations for the clinical concept instance *fever*.

4 Evaluation

In recent years, significant research effort has centered on the evaluation of ontologies and ontology-like lexical resources, with a smorgasbord of techniques available (Zhu et al., 2009; Brank et al., 2005). Yet no single evaluation method has achieved "best practice" status for all contexts. As our ontology is an application ontology designed to facilitate NLP in a highly constrained domain (that is, text analysis and information extraction from clinical reports) the notion of *coverage* is vital. There are two distinct questions here:

1. Can we map between the various textual instantiations of ILI concepts clinical reports and our ontology concepts? That is, are the NLP resources available in the ontology (keywords, regular expressions) adequate for the mapping task?

2. Do we have the right ILI concepts in our ontology? That is, do we adequately represent all the ILI concepts that occur in clinical reports?

Inspired by Grigonyte et al. (2010), we attempted to address these two related issues using

[12]`http://code.google.com/p/ss-ontology`

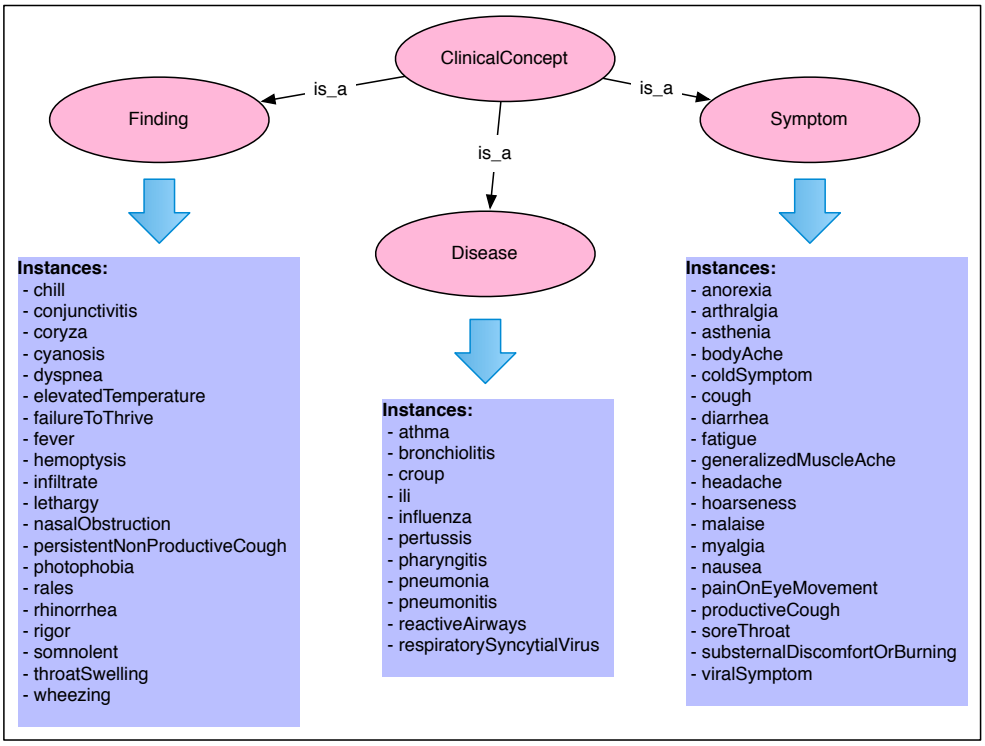

Figure 1: Clinical concepts.

techniques derived from terminology extraction and corpus linguistics. Our method consisted of assembling a corpus of twenty Emergency Room clinical reports which had been flagged by experts (not the current authors) as relevant to ILI. Note that these articles were not used in the initial knowledge engineering phase of the project. We then identified the "best" twenty five terms from these clinical reports using two tools, *Termine* and *KWExT*.

1. ***Termine*** (Frantzi et al., 2000) is a term extraction tool hosted by Manchester University's National Centre for Text Mining which can be accessed via web services.[13] It uses a method based on linguistic preprocessing and statistical methods. We extracted 231 terms from our twenty ILI documents (using *Termine*'s default configuration). Then we identified the twenty-five highest ranked *disease*, *finding* and *symptom* terms (that is, discarding terms like "hospital visit" and "chief complaint").

2. ***KWExT*** (Keyword Extraction Tool) (Conway, 2010) is a Linux based statistical keyword extraction tool.[14] We used *KWExT* to extract 1536 unigrams, bigrams and trigrams using the log-likelihood method (Dunning, 1993). The log-likelihood method is designed to identify n-grams that occur with the most frequency compared to some reference corpus. We used the FLOB corpus,[15] a one million multi-genre corpus consisting of American English from the early 1990s as our reference corpus. We ranked all n-grams according to their statistical significance and then manually identified the twenty-five highest ranked *disease*, *finding* and *symptom* terms.

Term lists derived using the *Termine* and *KWExT* tools are presented in Tables 1 and 2 respectively. For both tables, column two ("Term") details each of the twenty-five "best" terms (with respect to each term recognition algorithm) ex-

[13] www.nactem.ac.uk/software/termine/

[14] http://code.google.com/p/kwext/

[15] www.webcitation.org/5q1aKtnf3

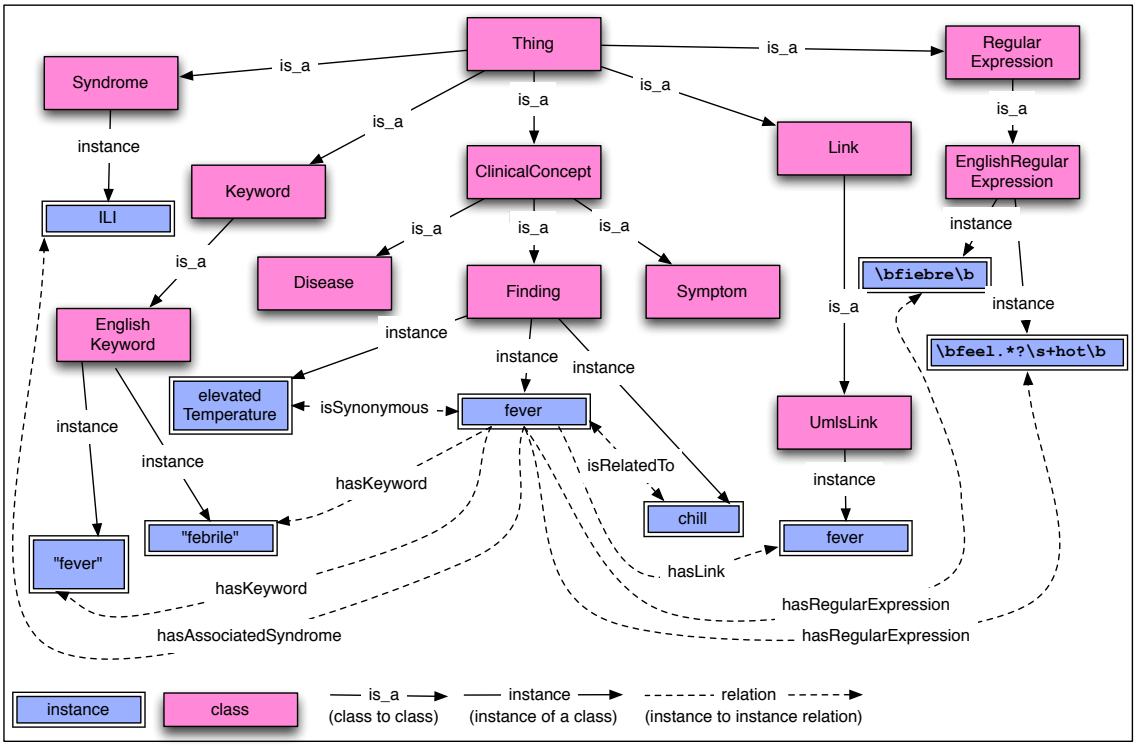

Figure 2: Example of clinical concept "fever" and its important relations (note the diagram is simplified).

tracted from our twenty document ILI corpus. Column three ("Concept") specifies the concept in our ontology to which the term maps (that is, the lexical resources in the ontology — keywords and regular expressions — can map the term in column two to the clinical concept in column three). For instance the extracted term *slight crackles* can be mapped to the clinical concept RALE using the keyword "crackles." Note that "-" in column three indicates that no mapping was possible. Underlined terms are those that *should* be mapped to concepts in the ontology, but currently are not (additional concepts and keywords will be added in the next iteration of the ontology).

There are two ways that mappings can fail here (mirroring the two questions posed at the beginning of this section). "Shortness of breath" *should* map to the concept DYSPNEA, but there is no keyword or regular expression that can bridge between text and concept. For the terms "edema" and "lymphadenopathy" however, no suitable candidate concept exists in the ontology.

5 Further Work

While the current ontology covers only ILI, we have firm plans to extend the current work along several different dimensions:

- Developing new relations, to include modeling DISEASE → SYMPTOM, and DISEASE → FINDING relations (for example TONSILLITIS **`hasSymptom`** SORE THROAT).
- Extend the application ontology beyond ILI to several other syndromes of interest to the biosurveillance community. These include:
  - *Rash Syndrome*
  - *Hemorrhagic Syndrome*
  - *Botulic Syndrome*
  - *Neurological Syndrome*
- Currently, we have links to UMLS (and also Snomed-CT and BioCaster). We intend to extend our coverage to the MeSH vocabulary (to facilitate mining PubMed) and also the Infectious Disease Ontology.

	Term	Concept
1	abdominal pain	-
2	chest pain	-
3	urinary tract infection	-
4	sore throat	SORE THROAT
5	renal disease	-
6	runny nose	CORYZA
7	body ache	MYALGIA
8	respiratory distress	PNEUMONIA
9	neck stiffness	-
10	yellow sputum	-
11	mild dementia	-
12	copd	-
13	viral syndrome	VIRAL SYN.
14	influenza	INFLUENZA
15	febrile illness	FEVER
16	lung problem	-
17	atrial fibrillation	-
18	severe copd	-
19	mild cough	COUGH
20	asthmatic bronchitis	BRONCHIOLITIS
21	coronary disease	-
22	dry cough	COUGH
23	neck pain	-
24	bronchial pneumonia	PNEUMONIA
25	slight crackles	RALE

Table 1: Terms generated using the *Termine* tool

	Term	Concept
1	cough	COUGH
2	fever	FEVER
3	pain	-
4	shortness of breath	-
5	vomiting	-
6	influenza	INFLUENZA
7	pneumonia	PNEUMONIA
8	diarrhea	DIARRHEA
9	nausea	NAUSEA
10	chills	CHILL
11	abdominal pain	-
12	chest pain	-
13	edema	-
14	cyanosis	CYANOSIS
15	lymphadenopathy	-
16	dysuria	-
17	dementia	-
18	urinary tract inf	-
19	sore throat	SORE THROAT
20	wheezing	WHEEZING
21	rhonchi	-
22	bronchitis	BRONCHIOLITIS
23	hypertension	-
24	tachycardia	-
25	respiratory distress	PNEUMONIA

Table 2: Terms generated using the *KWExT* tool

- Currently evaluation strategies have concentrated on *coverage*. We plan to extend our auditing to encompass both *intrinsic* evaluation (for example, have our relations evaluated by external health professionals using some variant of the "laddering" technique (Bright et al., 2009)) and *extrinsic* evaluation (for example, plugging the application ontology into an NLP pipeline for Named Entity Recognition and evaluating its performance in comparison to other techniques).

In addition to these ontology development and evaluation goals, we intend to use the ontology as a "gold standard" against which to evaluate automatic term recognition and taxonomy construction techniques for the syndromic surveillance domain. Further, we seek to integrate the resulting ontology with the BioCaster ontology allowing the potential for limited interlingual processing in priority languages (in the United States, Spanish).

Currently we are considering two ontology integration strategies. First, using the existing mappings we have created between the ILI ontology and BioCaster to access multi-lingual information (using OWL datatype properties). Second, fully integrating — that is, *merging* — the two ontologies and creating object property relations between them.

For example (using strategy 1), we could move from the string "flu" in a clinical report (identified by the `\bflu\b` regular expression) to the ILI ontology concept `ili:influenza`. In turn, `ili:influenza` could be linked (using a datatype property) to the BioCaster root term `biocaster:DISEASE_378` (which has the label "Influenza (Human)."). From the BioCaster root term, we can — for example — generate the translation "Gripe (Humano)" (Spanish).

6 Conclusion

The ILI application ontology developed from the need for knowledge resources for the text mining of clinical documents (specifically, Emergency Room clinical reports). Our initial evaluation indicates that we have good coverage of our domain, although we plan to incrementally work on improving any gaps in coverage through a process of active and regular updating. We have described our future plans to extend the ontology to new syndromes in order to provide a general commu-

nity resource to facilitate data sharing and integration in the NLP based syndromic surveillance domain. Finally, we actively solicit feedback on the design, scope and accuracy of the ontology.

Acknowledgments

This project was partially funded by Grant Number 3-R01-LM009427-02 (NLM) from the United States National Institute of Health.

References

Brank, J., Grobelnik, M., and Mladenić, D. (2005). A Survey of Ontology Evaluation Techniques. In *Proceedings of the Conference on Data Mining and Data Warehouses (SiKDD 2005)*, pages 166–170.

Bright, T., Furuya, E., Kuperman, G., and Bakken, S. (2009). Laddering as a Technique for Ontology Evaluation. In *American Medical Informatics Symposium (AMIA 2009)*.

Chen, H., Zeng, D., and Dang, Y. (2010). *Infectious Disease Informatics: Syndromic Surveillance for Public Health and Bio-Defense*. Springer, New York.

Collier, N., Shigematsu, M., Dien, D., Berrero, R., Takeuchi, K., and Kawtrakul, A. (2006). A Multilingual Ontology for Infectious Disease Surveillance: Rationale, Design and Challenges. *Language Resources and Evaluation*, 40(3):405–413.

Conway, M. (2010). Mining a Corpus of Biographical Texts Using Keywords. *Literary and Linguistic Computing*, 25(1):23–35.

Cooper, D. (2007). *Disease Surveillance: A Public Health Informatics Approach*, chapter Case Study: Use of Tele-health Data for Syndromic Surveillance in England and Wales, pages 335–365. Wiley, New York.

Doyle, T., Ma, H., Groseclose, S., and Hopkins, R. (2005). PHSkb: A Knowledgebase to Support Notifiable Disease Surveillance. *BMC Med Inform Decis Mak*, 5:27.

Dunning, T. (1993). Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics*, 19(1):61–74.

Eysenbach, G. (2006). Infodemiology: Tracking Flu-Related Searches on the Web for Syndromic Surveillance. In *American Medical Informatics Association Annual Symposium Proceedings (AMIA 2006)*, pages 244–248.

Frantzi, K., Ananiadou, S., and Mima, H. (2000). Automatic Recognition for Multi-word Terms. *International Journal of Digital Libraries*, 3(2):117–132.

Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., and Schneider, L. (2002). Sweetening Ontologies with DOLCE. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, pages 166–181.

Grigonyte, G., Brochhausen, M., Martin, L., Tsiknakis, M., and Haller, J. (2010). Evaluating Ontologies with NLP-Based Terminologies - A Case Study on ACGT and its Master Ontology. In *Formal Ontology in Information Systems: Proceedings of the Sixth International Conference (FOIS 2010)*, pages 331–344.

Henning, K. (2004). What is Syndromic Surveillance? *MMWR Morb Mortal Wkly Rep*, 53 Suppl:5–11.

Lombardo, J., Burkom, H., Elbert, E., Magruder, S., Lewis, S. H., Loschen, W., Sari, J., Sniegoski, C., Wojcik, R., and Pavlin, J. (2003). A Systems Overview of the Electronic Surveillance System for the Early Notification of Community-Based Epidemics (ESSENCE II). *J Urban Health*, 80(2 Suppl 1):32–42.

Miller, G. (1995). WordNet: A Lexical Database for English. *Communications of the Association for Computing Machinary*, 38(11):39–41.

Niles, I. and Pease, A. (2003). Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology. In *Proceedings of the 2003 International Conference on Information and Knowledge Engineering (IKE 03)*, pages 23–26.

Okhmatovskaia, A., Chapman, W., Collier, N., Espino, J., and Buckeridge, D. (2009). SSO: The Syndromic Surveillance Ontology. In *Proceedings of the International Society for Disease Surveillance*.

Scholer, M. (2004). Development of a Syndrome Definition for Influenza-Like-Illness. In *Proceedings of American Public Health Association Meeting (APHA 2004)*.

Tsui, F., Espino, J., Dato, V., Gesteland, P., Hutman, J., and Wagner, M. (2003). Technical Description of RODS: a Real-Time Public Health Surveillance System. *J Am Med Inform Assoc*, 10(5):399–408.

Wagner, M., Gresham, L., and Dato, V. (2006). *Handbook of Biosurveillance*, chapter Case Detection, Outbreak Detection, and Outbreak Characterization, pages 27–50. Elsevier Academic Press.

Zhu, X., Fan, J.-W., Baorto, D., Weng, C., and Cimino, J. (2009). A Review of Auditing Methods Applied to the Content of Controlled Biomedical Terminologies. *Journal of Biomedical Informatics*, 42(3):413 – 425.

Interfacing the Lexicon and the Ontology in a Semantic Analyzer

Igor Boguslavsky
Universidad Politécnica de Madrid
Institute for Information Transmission Problems of the Russian Academy of Sciences
igor.m.boguslavsky@gmail.com

Leonid Iomdin
Institute for Information Transmission Problems of the Russian Academy of Sciences
iomdin@iitp.ru

Victor Sizov
Institute for Information Transmission Problems of the Russian Academy of Sciences
sizov@iitp.ru

Svetlana Timoshenko
Institute for Information Transmission Problems of the Russian Academy of Sciences
nyrestein@gmail.com

Abstract

We discuss the possibility to link the lexicon of an NLP system with a formal ontology in an attempt to construct a semantic analyzer of natural language texts. The work is carried out on the material of sports news published in Russian media.

1 Introduction

Many Semantic Web applications need a much deeper semantic analysis of the text than is used today. Not only should the ontology elements be extracted from the textual data but also it is important to interpret the text in terms of the ontology. It is essential that IE and QA systems should be able to discover semantic similarity between the texts if they express the meaning in different ways. Cf. synonymous sentences (1) – (3):

(1) *Real Madrid and Barcelona will meet in the semi-finals on Thursday*.

(2) *The semi-final match between Real Madrid and Barcelona will take place on Thursday.*

(3) *The adversary of Real Madrid in the semi-finals on Thursday will be Barcelona.*

If we wish to extract the meaning from the text irrespective of the way it is conveyed, we should construct a semantic analyzer capable of producing identical semantic structures for sentences (1)-(3), or at least semantic structures whose equivalence can be easily demonstrated.

The problem becomes much more difficult if text understanding includes access to text-external world knowledge. For example, sentences (1)-(3) describe the same situation as (4).

(4) *The semi-finals on Thursday will see the champion of the UEFA Champions League 2008-2009 and the team of Manuel Pellegrini*.

To account for this synonymy, the system should know that it was the football club *Barcelona* who won the UEFA Champions League in 2008-2009, and that Manuel Pellegrini is the coach of *Real Madrid*. This implies that linguistic knowledge should be linked with ontological resources. The creation of a semantic analyzer of this type goes far beyond the task of assigning ontological classes to words occurring in the text. It requires a powerful wide-coverage linguistic processor capable of building coherent semantic structures, a knowledge-extensive lexicon, which contains different types of lexical information, an ontology, which describes objects in the domain and their properties, a repository of ground-level facts, and an inference engine.

Proceedings of the 6th Workshop on Ontologies and Lexical Resources (Ontolex 2010), pages 67–76,
Beijing, August 2010

A project NOVOFUT aiming at the development of a semantic analyzer of this type for Russian texts has started at the Institute for Information Transmission Problems of the Russian Academy of Sciences. It covers the domain of news about football. There are several reasons for this choice of domain. First, the news texts are written primarily for the general public, so that their understanding does not require specialized expert knowledge. This is a major advantage since it significantly facilitates the acquisition of the ontology. Second, the language typical of sports journalism is rich enough, which makes its interpretation linguistically non-trivial. Last but not least, sports enjoy enormous public interest. There are many sports portals publishing multifarious information on the daily (and sometimes hourly) basis and visited by a lot of people. Enhanced Question-Answering and Information Extraction in this domain are likely to attract many users.

The NOVOFUT semantic analyzer reuses many types of resources created or accumulated by the team in previous work. In this paper we focus on the static resources used by the analyzer – the lexicon and the ontology. The plan of the presentation is as follows. In Section 2 we discuss related work. In Section 3 we will briefly describe the linguistic processor we build on and its lexicon. Section 4 outlines a small-scale ontology developed for the project. The correlation between natural language words as presented in the lexicon and the ontology is the main concern of Section 5. In Section 6 the interface between the ontology and the lexicon is discussed. Future work is outlined in Section 7.

2 Related work

The link between the ontologies and NL texts is investigated in two directions – "from the ontology towards NL texts" and "from the texts towards the ontology". In the first case written texts are used as a means for ontology extension and population. To name but a few authors, McDowell and Cafarella (2006) start from an ontology and specify web searches that identify in the texts possible semantic instances, relations, and taxonomic information. In (Schutz and Buitelaar 2005) an interesting attempt is made to extract ontological relations from texts. (Buitelaar et al. 2008, Magnini et al. 2006, Maynard & al. 2006) are further advances in the direction of ontology population.

Finding NL equivalents to ontological elements and be monolingual or multilingual. A metamodel for linking conceptual knowledge with its lexicalizations in various languages is proposed in (Montiel-Ponsoda et al. 2007).

The second direction research starts from NL texts and tries to interpret them in terms of the ontology. In most cases, this takes the form of marking the text with ontological classes and instances. A typical example is (Sanfilippo et al. 2006). One should also mention the work based on the Generalized Upper Model (GUM), which is meant for interfacing between domain models and NL components (Bateman et al. 1995)

Our work belongs to this second direction, but our aim is not limited to finding ontological correlates to words. In many aspects we were inspired by the ontological semantic approach developed in the Mikrokosmos framework (cf. Nirenburg and Raskin 2004). We share many of its postulates and concrete solutions. In particular, semantic analysis of the text should be based on both linguistic and extra-linguistic knowledge. Linguistic knowledge is implemented in language grammars and dictionaries, while extra-linguistic knowledge is comprised in an ontology, which enumerates concepts, describes their properties and states interrelationships between them, and a fact repository which accumulates ground-level facts, such as, in our case, the data about concrete players, teams and matches. To a large extent, the ontology serves as the semantic language for meaning representation.

At the same time, there exist some differences between our approaches determined by the linguistic model adopted. Our work is based on the Meaning ⇔ Text theory (Mel'čuk 1974, 1996). In particular, we make extensive use of lexical functions, which constitute one of the prominent features of this theory. Thanks to lexical functions it turns out possible to reduce a wider range of synonymous sentences to the same semantic

structure, and in many cases, improve the performance of search engines (see e.g. Apresjan et al. 2009). Another difference between the Mikrokosmos approach and ours concerns the fact that the Mikrokosmos ontology is written in a specific in-house formalism. Our emphasis is on using as far as possible standard ontology languages (OWL, SWRL), in order to obtain interoperability with a wide and ever growing range of semantic web resources and inference engines.

3 The ETAP-3 Linguistic Processor and its Lexicon.

The multifunctional ETAP-3 linguistic processor, developed by the Computational Linguistics Laboratory of the Institute for Information Transmission Problems, Russian Academy of Sciences, Moscow, (see e.g. Apresjan et al. 2003), is the product of decades of research and development in the field of language modelling.

At the moment, ETAP-3 consists of a number of options, including

1) a rule-based machine translation system working both ways between Russian and English (plus several prototypes for other languages – French, German, Spanish, Korean and Arabic);

2) a system of synonymous and quasi-synonymous paraphrasing of sentences;

3) an environment for deep annotation of text corpora, in which SynTagRus, the only corpus of Russian texts tagged morphologically, syntactically (in the dependency tree formalism), and lexically was created, and

4) a Universal Networking Language (UNL) module, responsible for automatic translation of natural language text into a semantic interlingua, UNL, and the other way around.

The ETAP-3 processor is largely based on the general linguistic framework of the Meaning ⇔ Text theory by Mel'čuk. An important complement to this theory was furnished by the theory of systematic lexicography and integrated description of language proposed by Jurij Apresjan (2000).

One of the major resources used in ETAP-3 is **the combinatorial dictionary**. It offers ample and diverse data for each lexical entry. In particular, the entry may list the word's syntactic and semantic features, its subcategorization frames, as well as rules (or reference to rules) of a dozen types, which make it possible to describe peculiar behavior of individual words and exceptions to general rules in a complete and consistent way. Many dictionary entries contain information on **lexical functions,** to be discussed below in some detail.

The entry of the combinatorial dictionary has a number of zones, one of which provides the properties of the word that are manifested in the given language, while all the other zones contain information on the match between this word and its equivalent in a particular language. For example, the EN zone in the Russian combinatorial dictionary entry contains information on the translational equivalents of the respective Russian word into English. One field (TRANS) gives the default single-word translation (or several such translations) of this word in English. Other fields contain less trivial translation rules, or references to such rules.

A newly introduced ONTO zone offers information underlying the match between the Russian word and its counterparts in the ontology.

4 Ontology of football.

The ontology we are working with focuses in the first place on football. It contains information on teams, players, football field, sport events, and their properties. However, we want it to be extendable to other sports as well. That is why some classes are more general than would be needed for football alone. For example, instead of having one class `FootballPlayer`, the ontology has a more general class `Sportsman`, of which `FootballPlayer` is a subclass. An equivalence restriction states that `FootballPlayer` is a `Sportsman` whose SportType is `football`. In this way, sportsmen doing different types of sports can be treated by the ontology in a uniform way.

The football ontology is written in SWRL, which is OWL augmented with rules (Horrocks et al. 2004). In compiling it, we used some existing ontologies dealing with foot-

ball (http://www.lgi2p.ema.fr/~ranwezs/ontologies/soccerV2.0.daml). As usual, properties of classes are inherited by the subclasses. For example, the `Match` class is a subclass of `SportEvent`, which in its turn is a subclass of `Event`. `Match` inherits from `Event` the properties of having definite `Time` and `Place`. From `SportEvent` it inherits the fact that its participants should be `Sport-Agents`. Its own properties are: the number of participants is 2 (as opposed to championships, which have more) and it has a definite sport type (as opposed to Olympics, which involve many sport types). A subclass of `Match` is `Derby`, in which both participants should be from the same city or region. This property is implemented by means of a SWRL rule (cf. below). Another rule assigned to `Match` states that if its sport type is football (or any other team sport), then its participants should be teams and not individual sportsmen, as is the case in tennis or chess. `Sportsman` is a subclass of two classes: `Person`, from which it inherits the property of having a name and a birth date, and `SportAgent`, which includes also `Team` and from which it inherits the property of having a definite sport type and a coach.

5 Correlation between the words and the elements of the ontology.

As mentioned above, the ontology plays a two-fold role. On the one hand, it is a repository of domain-specific knowledge, and on the other hand, it is a semantic metalanguage used for representing the meaning of natural language texts. All meaningful natural languages elements (words, grammatical constructions, morphological features) must be interpreted in ontological terms. This makes the correlation between the lexicon and the ontology far from trivial. In this section, we will present several typical situations and illustrate them with examples.

5.1 One-to-one correspondence between NL words and ontology elements.

The simplest situation occurs when a word directly corresponds to an ontology element – a class, an individual, a relation, an attribute or its value. Example:

(5) *Real Madrid pobedil Arsenal so sčetom 3:1* 'Real Madrid defeated Arsenal 3 to 1'.

Here *Real Madrid* and *Arsenal* are individuals – instances of the `Team` class, the verb *to defeat* corresponds to the `WinEvent` class, and numbers *3* and *1* are values of attributes `scoreWinner` and `scoreLoser`, respectively. In the semantic structure (SemS) classes are represented by instances supplied by a unique numeric identifier. SemS for sentence (5) looks as follows:

```
hasWinner(WinEvent01, Real-
Madrid)&hasLoser(WinEvent01,
Arsenal)& scoreWinner
(WinEvent01,3)&
scoreLoser(WinEvent01,1)
```

5.2 One ontology element – several words ("multi-word concepts").

This is a very typical situation, especially in the area of terminology. For example, in ordinary language *željtaja kartočka* 'a yellow card' is a free noun phrase that simply denotes a card whose colour is yellow. In the sports domain, it is a single concept that refers to one of several possible punishments a referee can mete out for a rules infraction. Therefore, it is represented as one element in the ontology. Some other examples of multi-word sport concepts: *uglovoj udar* 'corner', *svobodnyj udar* 'free kick', *pravyj poluzaščitnik* 'right tackle'.

5.3 One word – several ontological elements.

Many words that can be interpreted in terms of ontological elements do not correspond to any single class, relation or instance. Their definition consists in a configuration of these elements. Most often, it is a class with some of the properties instantiated. In principle, often there are two options: one can either postulate two different classes (e.g.

Sportsman and FootballPlayer as its subclass), or only one class (Sportsman) and represent the *football player* as a Sportsman whose SportType property is football. There is no general solution to this alternative. In general, it is desirable to obtain a parsimonious ontology and refrain from introducing new classes, if one can define a concept in terms of existing classes and their properties. However, if a concept has important properties of its own, it is preferable to present it in the ontology as a separate class. In our example, FootballPlayer has Specialization which other sportsmen do not have (goalkeeper, forward, back, etc.) For this reason, it is postulated as a separate class of the ontology, with the indication of its equivalence to the anonymous class "Sportsman and hasSportType football".

An interesting and typical case are adjectival modifiers to nouns of the type *ispanskij* 'Spanish', *francuzskij* 'French', *moskovskij* 'of Moscow', and the like. Usually, dictionaries provide a very general definition of such words. For example, the COBUILD English dictionary gives only one meaning for the adjective *Spanish*: 'belonging or relating to Spain, or to its people, language or culture'. However, in real life situations this word is often interpreted by the speakers in a more specific way, according to the context. Ontological description should try, as far as possible, to take contextual factors into account and make explicit the specific interpretation of the modifier in each particular case. Sometimes, it can be done by means of rules that refer to ontological categories. For example, the meaning of the adjective *ispanskij* 'Spanish' mentioned above, when applied to geographical objects (*rivers, cities, mountains, roads,* etc.), narrows down to (hasLocation Spain). If this adjective modifies a noun denoting an industrial or agricultural product (*car, wine, olive oil,* etc.), it is rather interpreted as (producedIn Spain). We will hardly understand the phrase *Spanish wine* as denoting the wine located in Spain. Textual objects (*songs, literature, poetry,* etc.) move the adjective towards denoting the Spanish language: (inLanguage Spanish). Of course, these rules are not always sufficient for disambiguation. If an object falls into more than one category, several interpretations are possible. In particular, a book is both a textual and a consumer object. Therefore, *a Spanish book* can be interpreted as a book written in Spanish, and as a book published in Spain.

In many cases, adjectives serve as arguments of nouns. The semantic role of this argument may be different for different nouns (cf. (6-8)), and even for the same noun (cf. (9-11)):

(6) *presidential decree* – 'the president issued a decree': hasAgent(decree,president);

(7) *presidential elections* – 'somebody elects the president': hasObject(elect,president);

(8) *Babylonian invasion* – 'Babylon invaded some city or country': hasAgent(invade,Babylon); but not 'some city or country invaded Babylon': hasObject(invade,Babylon);

(9) *economic advisor* – 'advises in the area of economics': hasTopic(advisor,economics);

(10) *American advisor*: hasNationality(advisor,USA);

(11) *presidential advisor* – 'advises to the president': hasAddressee(advisor,president).

5.4 A word is interpreted in ontological terms but does not have any fixed ontological equivalent.

There is a large class of words that denote individuals which are in a certain relation to other individuals: *brother, sister, uncle, wife, friend, enemy, classmate, co-author, co-regent, coeval, adversary, ally,* etc. Of course, these words can be easily represented as ontology properties: hasBrother(John, Bill), hasSister(John, Mary). However, such representation does not reveal the meaning of the concepts. Being a brother of somebody means being a male and having common parents with this person. This meaning shares the second component ('having common parents') with the property of being a sister and

differs from it in the first component ('being a male'). Such a definition of meanings requires the use of variables. This is the point where the OWL expressive capacity is insufficient and one has to recur to SWRL rules:

```
Person(?person1)&Gender (?person1,male)&hasParent(?person1, ?person3) &hasParent (?person2,?person3)→ brother(?person1, ?person2)

Person(?person1)&Gender (?person1,female)&hasParent(?person1, ?person3)& hasParent (?person2,?person3)→ sister(?person1,?person2)
```

In a similar way one can define the concept of *adversary* (in sports), as used for example in sentence (3) above. *Adversary of Z* is someone different from Z who plays in the same match as Z:

```
SportAgent(?agent)& Match(?match)& hasParticipant(?match,?agent)& hasParticipant(?match,?z)& differentFrom(?agent,?z) → adversary(?agent,?z)
```

Among the words that require variables for their ontological definition are not only relational nouns. There are many other words that cannot be translated into ontological categories without claiming identity (or difference) of the properties of some individuals. Here are some examples from the football domain.

A *derby* is a match whose participants are from the same city or region. Our ontology defines the concept of *derby* as follows:

```
hasParticipant(?match, ?participant1)& hasParticipant (?match, ?participant2)&differentFrom (?participant1,?participant2) &hasLocation (?participant1, ?location) &hasLocation (?participant2,?location) → derby(?match)
```

Pobednyj gol ('decisive goal') is a goal which was scored when both teams had equal score and which was the last goal in the match. However, since having no subsequent goals cannot be expressed in SWRL we will convey this idea by saying that the goal brought the victory in the match to one of the teams. We will need the following classes and properties:

`GoalEvent`, with the properties: `hasAgent`, `atMinute`, e.g. *on the tenth minute*, `inMatch`, `hasResult` (inherited from the more general class `Event`).

`SituationInMatch` (the score at a given moment), with the properties: `inMatch`, `atMinute`, `scoreParticipant1`, `scoreParticipant2`.

`WinEvent`, with the properties: `hasWinner`, `hasLoser`.

`Team`, with the property `hasPart`, to be filled by instances of `Sportsman`.

Besides that, we need the property `timeImmediatelyBefore`, inherited by moments of time from `Time`.

We will describe the situation by means of two rules. Rule (12) says that the goal that brought a victory can be called a decisive goal. Rule (13) complements this description by saying that if a goal brings a victory, the winner is the team whose player scored it and this goal was scored at the moment when both teams had equal score.

(12)
```
GoalEvent(?goal)&WinEvent (?victory)& hasResult(?goal,?victory) → decisiveGoal(?goal)
```

(13)
```
hasResult(?goal,?victory) &hasAgent(?goal,?player)& hasPart (?team,?player)&atMinute (?goal,?min0)&inMatch (?goal, ?match)& timeImmediatelyBefore(?min1,?min0)& SituationInMatch (?situation)&inMatch (?situation,?match)& atMinute (?situation,?min1) → hasWinner(?victory,?team) &scoreParticipant1(?situation, ?n) &scoreParticipant2(?situation,?n).
```

5.5 Ontology and Lexical Functions.

A lexical function (LF), in the Meaning ⇔ Text theory (Mel'čuk 1996), has the basic properties of a multi-value mathematical

function. A prototypical LF is a triple of elements {***R,*** X, Y}, where ***R*** is a certain general semantic relation obtaining between the argument lexeme X (the keyword) and some other lexeme Y which is the value of ***R*** with regard to X (by a lexeme in this context we mean either a word in one of its lexical meanings or some other lexical unit, such as a set expression). Here are some examples for the $Oper_1$ and $Oper_2$ functions: $Oper_1$ (*control*) = *exercise* (*control*), $Oper_1$ (*research*) = *do* (*research*), $Oper_1$ (*invitation*) = *issue* (*an invitation*), $Oper_1$ (*doubt*) = *have* (*doubts*), $Oper_1$ (*defeat*) = *suffer* (*a defeat*), $Oper_1$ (*victory*) = *gain* (*a victory*), $Oper_1$ (*campaign*) = *wage (a campaign),* $Oper_2$ (*control*) = *be under* (*control*), $Oper_2$ (*analysis*) = *undergo* (*an analysis*), $Oper_2$ (*invitation*) = *receive* (*an invitation*), $Oper_2$ (*resistance*) = *encounter* (*resistance*), $Oper_2$ (*respect*) = *enjoy* (*respect*), $Oper_2$ (*obstacle*) = *face (an obstacle)*.

Y is often represented by a set of synonymous lexemes Y_1, Y_2, …., Y_n, all of them being the values of the given LF ***R*** with regard to X; e. g., Magn (*desire*) = *strong / keen / intense / fervent / ardent / overwhelming*. All the LF exponents for each word are listed in the lexicon.

LFs have a strong potential for advanced NLP applications. Apresjan *et al.* (2007) shows how LFs can be used in parsing, machine translation, paraphrasing. In parsing, LFs are used to resolve or reduce syntactic and lexical ambiguity. The MT system resorts to LFs to provide idiomatic target language equivalents for source sentences in which both the argument and the value of the same LF are present. The system of paraphrasing automatically produces one or several synonymous transforms for a given sentence or phrase by means of universal LF-axioms; for example: *He respects* [X] *his teachers – He has* [$Oper_1(S_0 (X))$] *respect* [S_0 (X)] *for his teachers – He treats* [Labor1-2(S_0 (X))] *his teachers with respect – His teachers enjoy* [$Oper_2(S_0(X))$] *his respect*. It can be used in a number of advanced NLP applications ranging from machine translation to authoring and text planning.

In ontologically-oriented semantic analysis different LFs are reflected in different ways.

An LF corresponds to an ontological class.

Many LFs represent bundles of words that are semantically identical or very close and therefore can serve as representatives of this common meaning. We illustrate this with two closely related LFs (Apresjan *et al*. 2008).

The meaning covered by $LiquFunc_0$ is 'to cause to cease to exist or to be taking place'. This concept corresponds, in particular, to the following English verbs: *to stop (the aggression), to lift (the blockade), to dispel (the clouds), to demolish (the building), to disperse (the crowd), to avert (the danger), to cure (the disease), to close (the dispute), to break up (the family), to annul (the law), to dissolve (the parliament), to denounce (the treaty), to bridge (the gap).* Another LF of the Liqu family – $LiquFact_0$ – refers to a different kind of elimination. It means 'to cause to cease functioning according to its destination'. When somebody *closes the eyes*, they do not cease to exist, they only stop functioning. Some more examples: *shut down (the factory), stop (the car), land (the airplane), depose (the king), switch off (the lamp), neutralize (the poison), empty (the bucket).*

These LFs, along with several dozen others, play a significant role not only in text understanding and generation. They contribute in an interesting way to one of the crucial functions of ontologies – inference of implicit knowledge. Important inference rules can be easily formulated in terms of LFs: if the blockade is lifted (=$LiquFunc_0$), it does not exist any more. Another example of the LF-based inference (this time it is LF $Real_1$): *He fulfilled* (= $Real_1$) *the promise to buy a bicycle* → *He bought a bicycle.*

It should be emphasized that, given a lexicon which contains LF data (which is the case of our ETAP dictionary), the acquisition of this part of the ontology is straightforward.

An LF generates an ontological relation.

This case can be illustrated by support verbs of the Oper-Func-Labor family that attach one of the arguments to the noun. For example, in sentence *Father gave me an advice* the subject of the $Oper_1$-support verb *to give* (*father*) is the Agent of *advice*, while in *The proposal received much attention* the subject

of the $Oper_2$-support verb *to receive* (the proposal) is the Object of *attention*. Other examples of $Oper_1$ and $Oper_2$ were given in 5.5 above. Some examples of other LFs of this family:

$Func_1$: *(fear) possesses (somebody), (rumour) reaches (somebody), (the blame) falls on (somebody) / (the blame) lies with (somebody), (control) belongs to (somebody), (responsibility) rests with (somebody).*

$Func_2$: *(proposal) consists in (something), (criticism) bears upon (something), (revenge) falls upon (somebody).*

$Labor_{1-2}$: *keep (something) under (control), submit (something) to (analysis), meet (somebody) with (applause), put (somebody) under (arrest), hold (somebody) in (contempt), bring (something) into (comparison with something), take (something) into (consideration).*

An LF has no ontological correlate.

This is the case of $Func_0$. This LF neither denotes a concept, nor attaches an argument to a concept. It only duplicates the meaning of its keyword and has no correlate in the SemS. For example, in sentence (2) above the phrase *the match took place* (= $Func_0$) is only represented by the concept `Match`. Other examples of $Func_0$: *(the snow) falls, (the wind) blows, (the danger) exists, (the war) is on, (changes) occur.*

6 Lexicon ↔ Ontology interface.

For the purposes of semantic analysis, the Russian dictionary and the ontology are linked in the same way as dictionaries of different languages are linked in Machine Translation options of the ETAP-3 system. As noted in Section 2, if the system performs translations from language *L* to language *L′*, all dictionary entries of *L* contain a special zone (ZONE: L′) where all translation variants of the given word into *L′* are recorded. The semantic analysis option uses the ONTO zone of the Russian dictionary. In this zone, two types of information may be written:

- **Default translation**. This is a one-word equivalent of the given word, which is used if no translation rule is applicable.

For example, Russian *komanda* 'team' has the `Team` class as its ontological counterpart. This is written in the ontological zone of *komanda* as follows:

ZONE: ONTO

TRANS: `Team`

Names of ontological individuals are also often translated by default.

- **Translation rules.** A rule is written every time one needs to carry out an action which does not boil down to the default translation.

Let us give several examples of translation rules written in the ONTO zone of the Russian lexicon. We will not give their formal representation and restrict ourselves to explaining what they are doing in plain words.

Pobeditel' 'winner' is a SportAgent (i.e. a sportsman or a team) that won some contest:

```
SportAgent(?x)&WinEvent(?y)&
hasWinner(?y,?x).
```

Phrases of the type *komanda NN* 'team of NN' (where NN is a proper human name in the genitive case) are translated in four different ways depending on the ontological information assigned to NN.

(a) If NN is the name of a player, the phrase is represented as 'the team of which NN is a player': *komanda Arshavina* 'Arshavin's team' = `Team(?team) &hasPart(?team,Arshavin).`

(b) If NN is the name of a coach, the phrase is represented as 'the team of which NN is the coach': *komanda Pellegrini* 'Pellegrini's team' = `Team(?team) &hasCoach(?team,Pellegrini)`

(c) If NN is the name of a captain, the phrase is represented as 'the team of which NN is the captain': *komanda Iraneka* 'Iranek's team' = `Team(?team)& hasCaptain(?team,Iranek)`

(d) If NN is neither a player, nor a coach, nor a captain, the phrase is represented as 'the team of which NN is a fan': *komanda Ivana* 'Ivan's team' = `Team(?team)& hasFan(?team,Ivan)`

It is well-known that genitive noun phrases (or phrases "N1 of N2" in English) are very vague semantically, and their interpretation is very much dependent on the context. This example shows that even within the

part/whole interpretation such a phrase, paradoxically, has two opposite varieties: either N2 is part of N1, as in *the team of Arshavin/Arshavin's team*, or N1 is part of N2, as in *the leg of the table*.

The following examples involve the property `hasLocation`, which characterizes both sport events (*The match took place in Madrid*), and sport agents (*the Ukrainian sportsman, a London club*).

Frequently, a football match is played in a location, such that one of the teams is from that location while the other is not. This situation can be represented by the following SemS:

(14)

```
Match(?match)&hasLocation(?match,?place)&hasParticipant
(?match, ?team1)&hasParicipant(?match,?team2)&differentFrom(
?team1,?team2)&hasLocation(?team1,?place)&¬hasLocation(?team
2,?place)
```

In the natural language this situation can be viewed from different angles and denoted by different words.

Xozjaeva 'home team' denotes a team that plays a match in a place it is from, the adversary being from a different place. *Gosti* 'visitors' is a team that plays a match in a location different from the place it is from, the adversary being the home team. *Prinimat'* 'to receive' means to play a match being a home team, to host it. *Igrat' v gostjax* lit.'to play being guests' means to play a match away.

Although all these words correspond to the same situation (14), their translation rules cannot be identical. The rules should not only introduce SemS (14), but also assure correct amalgamation of this SemS with SemSs of other words. In particular, the rule for *prinimat'* 'receive' should guarantee that in (15) Real Madrid instantiates variable `?team1` of (14), and Barcelona – variable `?team2`.

(15) *Real Madrid prinimal Barcelonu* 'Real Madrid hosted Barcelona'

The rule for *gosti* 'visitors' should see to it that in (16) `hasWinner` property of `WinEvent` be filled by variable `?team2` of (14):

(16) *Gosti vyigrali 3:1* 'the visitors won 3 to 1'

This is assured due to marking `?team2` in the *gosti* 'visitors' rule as the head element of SemS (14). Naturally, in the *xozjaeva* 'home team' rule the same role is assigned to `?team1`.

7 Future work.

In the continuation, it is planned to enlarge both the ontology and ONTO zone of the Russian lexicon. We are investigating the possibility of merging our small football ontology with some existing larger upper level ontology. The difficult task will be to unify our semantic rules with the axioms of this ontology.

A second direction of our future activity is connected with another component of the semantic analyzer, which we did not touch upon in this paper. It is the set of semantic rules which are not incorporated into the lexicon due to their general character. This component also requires significant enhancement.

An important extension of this work consists in introducing an inference component based on the SWRL rules.

Acknowledgement

This study has received partial funding from the Russian Foundation for Humanities (grant No. 10-04-00040a), which is gratefully acknowledged.

References

Apresjan, Ju. D. Systematic Lexicography. Oxford University Press, 2000, XVIII p., 304 p.

Apresjan, Jury, I. Boguslavsky, L. Iomdin, A. Lazursky, V.Sannikov, V.Sizov, L.Tsinman. ETAP-3 Linguistic Processor: a Full-Fledged NLP Implementation of the MTT // *MTT 2003, First International Conference on Meaning – Text Theory (June 16-18 2003).* Paris: Ecole Normale Supérieure, 2003. P. 279-288.

Apresjan, Jury, Igor Boguslavsky, Leonid Iomdin and Leonid Tsinman. Lexical Functions in Actual NLP Applications // Selected Lexical and Grammatical Issues in the Meaning–Text Theory. In honour of Igor Mel'čuk. (Ed. by Leo Wanner). John Benjamins, Studies in Language Companion Series 84. 2007. P. 199-230.

Apresjan, Ju.D., P.V. Djachenko, A.V. Lazursky, L.L. Tsinman. O kompjuternom uchebnike russkogo jazyka. [On a computer textbook of Russian.] *Russkij jazyk v nauchnom osveshchenii*. 2008, No. 2 (14). P. 48-112.

Apresjan Ju., I. Boguslavsky, L.Iomdin, L.Cinman, S.Timoshenko. Semantic Paraphrasing for Information Retrieval and Extraction. In: T.Andreasen, R.Yager, H.Bulskov, H.Christiansen, H.Legind Larsen (eds.) Flexible Query Answering Systems. Proceedings, 8th International Conference, 2009. Lecture Notes in Computer Science 5822. pp. 512-523

Bateman J., B. Magnini and G. Fabris. The Generalized Upper Model Knowledge Base: Organization and Use. In: Towards Very Large Knowledge Bases, pp. 60-72, IOS Press. 1995.

Buitelaar P., Ph. Cimiano, A.Frank, M. Hartung, S.Racioppa. (2008). "*Ontology-based Information Extraction and Integration from Heterogeneous Data Sources*." In: International Journal of Human-Computer Studies, 66(11).

Horrocks, I., P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof and M. Dean (2004). *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. // W3C Member Submission 21 May 2004.

Magnini B., Emanuele Pianta, Octavian Popescu, and Manuela Speranza. (2006). "*Ontology Population from Textual Mentions: Task Definition and Benchmark.*" In: Proceedings of the Ontology Population and Learning Workshop at ACL/Coling 2006.

Maynard D., Wim Peters, and Yaoyong Li. (2006). "*Metrics for Evaluation of Ontology-based Information Extraction.*" In: WWW 2006 Workshop on Evaluation of Ontologies for the Web (EON 2006)

McDowell Luke K., Michael Cafarella. Ontology-driven Information Extraction with OntoSyphon. In: Proceedings of the 5th International Semantic Web Conference (ISWC 2006). Volume 4273 of LNCS., Athens, GA, Springer (2006) 428 – 444

Mel'čuk, Igor. *Opyt teorii lingvisiticheskix modelej «Smysl ↔ Text»* [The theory of linguistic models of the Meaning ↔ Text type]. Moscow, 1974 (2nd edition 1999).

Mel'čuk, Igor. Lexical Functions: A Tool for the Description of Lexical Relations in Lexicon. *L. Wanner (ed.), Lexical Functions in Lexicography and Natural Language Processing*. Amsterdam, 1996, 37-102.

Montiel-Ponsoda, E., Aguado de Cea, G. y Gómez-Pérez, A. 2007 "Localizing ontologies in OWL. *From Text to Knowledge: The Lexicon/Ontology Interface. WS 2. The 6th International Semantic Web Conference*."

Nirenburg, Sergei, and Victor Raskin. *Ontological Semantics.* The MIT Press. Cambridge, Massachusetts. London, England, 2004.

Sanfilippo A., Tratz S., Gregory M., Chappell A., Whitney P., Posse Ch., Paulson P., Baddeley B., Hohimer R., White A. Automating Ontological Annotation with WordNet. In: Sojka P., Key-Sun Choi, Ch. Fellbaum, P. Vossen (Eds.): GWC 2006, Proceedings, pp. 85–93.

Schutz, A. and P. Buitelaar. RelExt: A Tool for Relation Extraction from Text in Ontology Extension. In: Y. Gil et al. (Eds.), ISWC 2005, LNCS 3729, pp. 593–606, 2005.

Ontolexical resources for feature based opinion mining : a case-study

Anaïs Cadilhac
IRIT
Toulouse University
cadilhac@irit.fr

Farah Benamara
IRIT
Toulouse University
benamara@irit.fr

Nathalie Aussenac-Gilles
IRIT
Toulouse University
aussenac@irit.fr

Abstract

Opinion mining is a growing research area both at the natural language processing and the information retrieval communities. Companies, politicians, as well as customers need powerful tools to track opinions, sentiments, judgments and beliefs that people may express in blogs, reviews, audios and videos data regarding a product/service/person/organisation/etc. This work describes our contribution to feature based opinion mining where opinions expressed towards each feature of an object or a product are extracted and summarized. The state of the art has shown that the hierarchical organization of features is a key step. In this context, our goal is to study the role of a domain ontology to structure and extract object features as well as to produce a comprehensive summary. This paper presents the developed system and the experiments we carried out on a case study: French restaurant reviews. Our results show that our approach outperforms standard baselines.

1 Introduction

Opinion mining is a growing research area both in natural language processing and information retrieval communities. Companies, politicians, as well as customers need powerful tools to track opinions, sentiments, judgments and beliefs that people may express in blogs, reviews, audios and videos data regarding a product/service/person/organisation/etc. The importance of emotion-oriented computing in the Web 2.0 has encouraged the creation of new search engines (like Tweetfeel (www.tweetfeel.com)) as well as the creation of a new research group within the W3C, namely the *Emotion Markup Language*, that aims to develop a representation language of the emotional states of a user or the emotional states to be simulated by a user interface. In addition, most information retrieval evaluation campaigns (TREC, NTCI, etc.) have already integrated an opinion track.

Computational approaches to sentiment analysis focus on extracting the affective content of a text from the detection of expressions of "bag of sentiment words" at different levels of granularity. These expressions are assigned a positive or a negative scalar value, representing a positive, a negative or neutral sentiment towards some topic. Roughly, research in this field can be grouped in four main categories (which are not exclusive):

- *Development of linguistic and cognitive models of opinion/sentiment* where already existing psycholinguistic theories of emotions are used to analyse how opinions are lexically expressed in texts (Wiebe et al, 2005; Read et al, 2007; Asher et al, 2009)
- *Elaboration of linguistic resources* where corpus based and dictionary based approaches are used to automatically or semi-automatically extract opinion bearing terms/expressions as well as their sentiment orientation (Strapparava et al., 2004; Turney and Littman, 2002)
- *Opinion extraction/analysis at the document* (Pang et al., 2002; Turney, 2002), *at the sentence or at the clause level* (Kim et al., 2006; Choi et al., 2005) where local

Proceedings of the 6th Workshop on Ontologies and Lexical Resources (Ontolex 2010), pages 77–86,
Beijing, August 2010

opinions are aggregated in order to compute the overall orientation of a document/sentence/clause.

- *Feature based opinion mining* (Hu and Liu, 2004; Popescu and Etzioni, 2005; Carenini et al., 2005; Cheng and Xu, 2008) where opinions expressed towards the features of an object or a product are exacted and summarized.

The work described in this paper feats into the last category. The aim is not to compute the general orientation of a document or a sentence, since a positive sentiment towards an object does not imply a positive sentiment towards all the aspects of this object, as in: I like this restaurant even if the service is slow. In feature based opinion mining, a holder (the person who posts the review) expresses a positive/negative or neutral opinions towards a main topic (the object or the product on which the holder expresses his opinions) and its associated features. As defined in (Hu and Liu, 2004), a feature can be a "part-of" of a topic (such as the screen of a camera) or a property of the "part-of" of the topic (such as the size of the screen). The expressed opinion can be explicit, as in "the screen of this camera is great", or implicit, as in "the camera is heavy", that expresses a negative opinion towards the weight of the camera. Same features can also be expressed differently, for example, "drink" and "beverage" refer to the same restaurant feature.

Having, for an object/product, the set of its associated features F={f1,…fn}, research in feature based opinion mining mostly focus on extracting the set F from reviews, and then, for each feature fi of F, extract the set of its associated opinion expressions OE={OE1,…OEj}. Once the set of couples (fi, OE) were extracted, a summary of the review is generally produced. During this process, the key questions are: how the set F of features can be obtained? How they are linguistically expressed? How they are related to each other ? Which knowledge representation model can be used to better organize product features and to produce a comprehensive summary?

To answer these questions, we propose in this paper to study the role of an ontology in feature based opinion mining. More precisely, our aim is to study how a domain ontology can be used to:

- *structure features:* we show that an ontology is more suitable than a simple hierarchy where features are grouped using only the "is-a" relation (Carenini et al., 2005; Blair-Goldensohn et al., 2008)
- *extract explicit and implicit features from texts*: we show how the lexical component as well as the set of properties of the ontology can help to extract, for each feature, the set of the associated opinion expressions.
- *produce a discourse based summary of the review:* we show how the ontology can guide the process of identifying the most relevant discourse relations that may hold between elementary discourse units.

The paper is organised as follows. We give in section 2, a state of the art of the main approaches used in the field as well as the motivations of our work. We present in the next section, our approach. Finally, in section 4, we describe the experiments we carried out on a case study: French restaurant reviews

2 Feature based Opinion mining

2.1 Related Works

Overall, two main families of work stand out: those that extract a simple list of features and those that organize them into a hierarchy using taxonomies or ontologys. The feature extraction process mainly concerns explicit features.

Works without knowledge representation models : The pioneer work in feature based opinion mining is probably the one of Hu and Liu (2004) that applies association rule mining algorithm to discover product features (nouns and noun-phrases). Heuristics (frequency of occurrence, proximity with opinion words, etc...) can eliminate irrelevant candidates. Opinion expressions (only adjective phrases) which are the closest to these features are extracted. A summary is then produced and displays, for each feature, both positive and negative phrases and the total number of these two categories.
To improve the feature extraction phase, Popescu and Etzioni (2005) suggest in their system

OPINE, to extract only nominal groups whose frequency is above a threshold determined experimentally using the calculation of PMI (Point-wise Mutual Information) between each of these nouns and meronymy expressions associated with the product. No summary is produced.
The main limitation of these approaches is that there are a great many extracted features and there is a lack of organization. Thus, similar features are not grouped together (for example, in restaurant domain, "*atmosphere*" and "*ambience"),* and possible relationships between features of an object are not recognized (for example, "*coffee"* is a specific term for "*drink*"). In addition, polarity analysis (positive, negative or neutral) of the document is done by assigning the dominant polarity of opinion words it contains (usually adjectives), regardless of polarities individually associated to each feature.

Works using feature taxonomies. Following works have a different approach: they do not look for a "basic list" of features but rather a list hierarchically organized through the use of taxonomies. We recall that a taxonomy is a list of terms organized hierarchically through specialization relationship type "is a sort of". Carenini et al. (2005) use predefined taxonomies and semantic similarity measures to automatically extract classic features of a product and calculate how close to predefined concepts in the taxonomy they are. This is reviewed by the user in order to insert missing concepts in the right place while avoiding duplication. The steps of identifying opinions and their polarity and the production of a summary are not detailed. This method was evaluated on the product review corpus of Hu and Liu (2004) and resulted in a significant reduction in the number of extracted features. However, this method is very dependent on the effectiveness of similarity measures used.
In their system PULSE, Gamon et al. (2005) analyze a large amount of text contained in a database. A taxonomy, including brands and models of cars, is automatically extracted from the database. Coupled with a classification technique, sentences corresponding to each leaf of the taxonomy are extracted. At the end of the process, a summary which can be more or less detailed is produced.
The system described in (Blair-Goldensohn et al., 2008) extracts information about services, aggregates the sentiments expressed on every aspect and produces a summary. The automatic feature extraction combines a dynamic method, where the different aspects of services are the most common nouns, and a static method, where a taxonomy grouping the concepts considered to be the most relevant by the user is used to manually annotate sentences. The results also showed that the use of a hierarchy significantly improves the quality of extracted features.

Works using ontologys. These works aim at organizing features using a more elaborated model of representation: an ontology Unlike taxonomy, ontology is not restricted to a hierarchical relationship between concepts, but can describe other types of paradigmatic relations such as synonymy, or more complex relationships such as composition relationship or space relationship.
Overall, extracted features correspond exclusively to terms contained in the ontology. The feature extraction phase is guided by a domain ontology, built manually (Zhao and Li, 2009), or semi-automatically (Feiguina, 2006; Cheng and Xu, 2008), which is then enriched by an automatic process of extraction / clustering of terms which corresponds to new feature identification.
To extract terms, Feiguina (2006) uses pattern extraction coupled to a terminology extractor trained over a set of features related to a product and identified manually in a few reviews. Same features are grouped together using semantic similarity measures. The system OMINE (Cheng and Xu, 2008) proposes a mechanism for ontology enrichment using a domain glossary which includes specific terms such as words of jargon, abbreviations and acronyms. Zhao and Li (2009) add to their ontology concepts using a corpus based method: sentences containing a combination of conjunction word and already recognized concept are extracted. This process is repeated iteratively until no new concepts are found.
Ontologys have also been used to support polarity mining. For example, (Chaovalit and Zhou, 2008) manually built an ontology for movie reviews and incorporated it into the polarity clas-

sification task which significantly improve performance over standard baseline.

2.2 Towards an ontology based opinion mining

Most of the researchers actually argue that the use of a hierarchy of features improves the performance of feature based opinion mining systems. However, works that actually use a domain ontology (cf. last section) exploit the ontology as a taxonomy using only the is-a relation between concepts. They do not really use all data stored in an ontology, such as the lexical components and other types of relations. In addition, in our knowledge, no work has investigated the use of an ontology to produce comprehensive summaries.

We think there is still room for improvement in the field of feature based sentiment analysis. To get an accurate appraisal of opinion in texts, it is important for NLP systems to go beyond explicit features and to propose a fine-grained analysis of opinions expressed towards each feature. Our intuition is that the full use of ontology would have several advantages in the domain of opinion mining to:

Structure features: ontologys are tools that provide a lot of semantic information. They help to define concepts, relationships and entities that describe a domain with unlimited number of terms. This set of terms can be a significant and valuable lexical resource for extracting explicit and implicit features. For example, in the following restaurant review: *cold and not tasty* the negative opinion *not tasty* is ambiguous since it is not associated to any lexicalised feature. However, if the term *cold* is stored in the ontology as a lexical realization of the concept *quality of the cuisine*, the opinion *not tasty* can be easily associated to the feature *cuisine* of the restaurant (note that the conjunction *and* plays an important role in the desambiguisation process). We discuss this point at the last section of the paper.

Extract features: ontologys provide structure for these features through their concept hierarchy but also their ability to define many relations linking these concepts. This is also a valuable resource for structuring the knowledge obtained during feature extraction task. In addition, the relations between concepts and lexical information can be used to extract implicit features. For example, if the concept *customer* is linked to the concept *restaurant* by the relation *to eat in,* a positive opinion towards the restaurant can be extracted from the review: *we eat well.* Similarly, if the concept *restaurant* is linked to the concept *landscape* with the relation *to view,* a positive opinion can be extracted towards *the look out of the restaurant* from the following review: *very good restaurant where you can savour excellent Gratin Dauphinois and admire the most beautiful peak of the Pyrénées*

Produce summaries. Finally, we also believe that ontologys can play a fundamental role to produce well organised summary and discursive representation of the review. We further detail this point at the last section of the paper.

3 Our approach

Our feature based opinion mining system needs three basic components: a lexical resource L of opinion expressions, a lexical ontology O where each concept and each property is associated to a set of labels that correspond to their linguistic realizations and a review R.

Following the idea described in (Asher et al, 2009), a review R is composed of a set of elementary discourse units (EDU). Using the discourse theory SDRT (Asher and Lascarides 2003) as our formal framework, an EDU is a clause containing at least one elementary opinion unit (EOU) or a sequence of clauses that together bear a rhetorical relation to a segment expressing an opinion. An EOU is an explicit opinion expression composed of a noun, an adjective or a verb with its possible modifiers (actually negation and adverb) as described in our lexicon L.

We have segmented conjoined NPs or APs into separate clauses—for instance, *the film is beautiful and powerful* is taken to express two segments: *the film is beautiful* and *the film is powerful*. Segments are then connected to each other using a small subset of "veridical" discourse relations, namely:

- *Contrast (a,b),* implies that a and b are both true but there is some defeasible implication

of one that is contradicted by the other. Possible markers can be *although, but.*

- *Result(a,b)* indicated by markers like *so, as a result*, indicates that the EDU b is a consequence or result of the EDU a.
- *Continuation(a,b)* corresponds to a series of speeches in which there are no time constraints and where segments form part of a larger thematic. For example, *"The average life expectancy in France is 81 years. In Andorra, it reaches over 83 years. In Swaziland it does not exceed 85 years."*
- *Elaboration(a,b)* describes global information that was stated previously with more specific information. For example, *"Yesterday, I spent a wonderful day. I lounged in the sun all morning. I ate in a nice little restaurant. Then at night, I met my friend Emily."*

In a review R, an opinion holder *h* comments on a subset S of the features of an object/product using some opinion expressions. Each feature corresponds to the set of linguistic realizations of a concept or a property of the domain ontology O. For example, in the following product review, EDUs are between square brackets, EOUs are between embraces whereas object features are underlined. There is a contrast relation between the EDU_b and EDU_c which makes up the opinion expressed within the EDU_d.

[I bought the product yesterday] $_a$. [*Even if the product is {excellent}*]$_b$, *[the design and the size are {very basic}*] $_c$, [*which is {disappointing} in this brand*] $_c$.

The figure below gives an overview of our system. First, each review R is parsed using the French syntactic parser Cordial[1], which provides, for each sentence, its POS tagging and the set of dependency relations. The review is then segmented in EDUs using the discourse parser described in (Afantenos and al, 2010).

For each EDU, the system :

1. Extracts EOUs using a rule based approach
2. Extracts features that correspond to the process of term extraction using the domain ontology

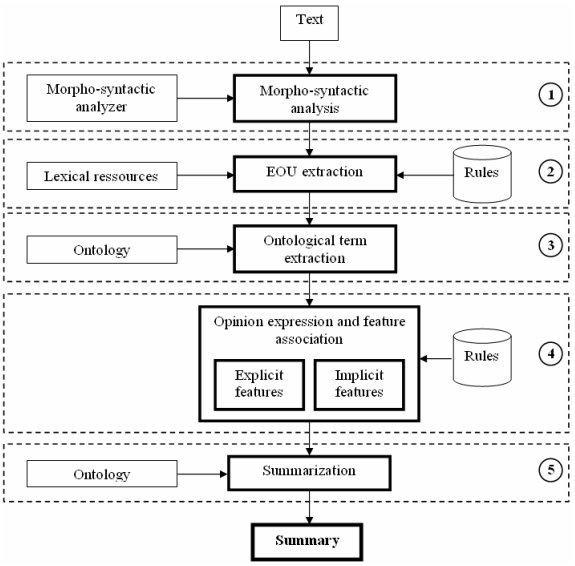

Figure 1 Overview of our system.

3. Associates, for each feature within an EDU, the set of opinion expressions
4. Produces a discourse based summary.

Since the summarization module is not done yet, we detail below the three first steps.

3.1 Extracting Elementary Opinion Units

We recall that an EOU is the smallest opinion unit within an EDU. It is composed of one and only one opinion word (a noun, an adjective or a verb) possibly associated with some modifiers like negation words and adverbs. For example, "*really not good*" is an EOU. An EOU can also be simply an adverb as in *too spicy*. Adverbs are also used to update our opinion lexicon, as in *too chic* where the opinion word *chic* is added. Finally, we also extract expressions of recommendation, such as : *go to this restaurant, you will not regret it*, which are very frequent in reviews.

3.2 Extracting features

This step aims at extracting for the review all the labels of the ontology. Since each concept and its associated lexical realizations correspond to explicit features, we simply project the lexical component of the ontology in the review in order to get, for each EDU, the set of features F. Of course, since our lexical ontology does not

[1] http://www.synapse-fr.com/Cordial_Analyseur/

cover all the linguistic realizations of concepts and properties in a given domain, many terms in the review can be missed. We show, in the next section, that linking features to opinion expressions can partially solve this problem.

To extract implicit features, ontology properties are used. We recall that these properties define relations between concepts of the ontology. For example, the property "*look at*" links "*customer*" and "*design*" concepts.

3.3 Associating opinions expressions to extracted features

In this step, the extracted opinion expressions in step 1 have to be linked to the features extracted in step 2 i.e. we have to associate to each EDU_i the set of couples (f_i, OE_i). During this step, we distinguish the following cases :

Case 1. *Known features and known opinion words.* For example, if the lexicon contains the words *really, good* and *excellent* and the ontology contains the terms *eating place* and *food* as a linguistic realization of the concepts *restaurant* and *food*, then this step allows the extraction from the EDU "*really good restaurant with excellent food''* the couples *(restaurant, really good)* and *(food, excellent)*. This example is quite simple but in many cases, features and opinion words are not close to each other which make the link difficult to find. Actually, our system deals with conjunctions (including comas) as in: *"I recommend pizzas and ice creams", "very good restaurant but very expensive"*

Case 2. *Known features and unknown opinion expressions,* as in the EDU "*acceptable prices*" where the opinion word *acceptable* has not been extracted in step 1 (cf. section 3.1). In this case, the opinion lexicon can be automatically updated with the retrieved opinion word.

Case 3. *Unknown features and known opinion expressions,* as in the EDU "*old fashion restaurant*" where the features *fashion* has not been extracted in step 2 (cf. section 3.2). In this case, the domain ontology can be updated by adding a new label to an existing concept or property or by adding a new concept or a new property in the right place to the ontology. However, since a user may express an opinion on different objects within a review, this step has to be done carefully. To avoid errors, we propose to manually update the ontology.

Case 4. *Opinion expressions alone,* as in the EDU "*It's slow, cold and not good*". This kind of EDU expresses an implicit feature. In this case, we use the ontology properties in order to retrieve the associated concept in the ontology. For example, in the sentence "*we eat very well*", the property "*eat*" of the ontology which links "*customer*" and "*food*" will allow the system to determine that "*very well*" refers to "*food*".

Case 5. *Features alone,* as in the EDU: "*Nice surrounding on sunny days with terrace*", even if the feature "*terrace*" is not associated to any opinion word, it is important to extract this information because it gives a positive opinion towards the restaurant. An EDU with features alone can also be an indicator of the presence of an implicit opinion expression towards the feature as in *this restaurant is a nest of tourists*

Actually, our system deals with all these cases except the last one.

4 Case study : mining restaurant reviews

In this section, we present the experiments we carried out on a case study: French restaurant reviews.

4.1 Corpus

For our experiments, we use a corpus of 58 restaurant reviews (40 positive reviews and 18 negatives reviews, for a total of 4000 words) extracted from the web site Qype[2]. Each review contains around 70 words and is composed of free comments on restaurants (but also on other objects like pubs, cinemas, etc.) with a lot of typos and syntactic errors. Each review appears in the web site with additional information such as the date of the review, the user name of the holder and a global rate from 1 (bad review) to 5 (very good review). In this experiment, we only use the textual comments posted. Figure 2 shows an example of a review form our corpus.

[2] http://www.qype.fr

<Start> Un bon petit resto sympa, près du centre, lumineux et à la déco sympa. Le service est de qualité, rapide et on y mange sain et bon. Je recommande ! <Stop>

Figure 2. Example of a restaurant review

4.2 Ontology

Since our aim is to study the role of a domain ontology to feature based opinion mining, we choose to reuse an existing ontology. However, for the restaurant domain, we do not find any public available ontology for French. We thus use a pre-existent ontology[3] for English as a basis coupled with additional information that we gather from several web sites[4]. We first translate the existing ontology to French and then adapt it to our application by manually re-organize, add and delete concepts in order to describe important restaurant features. Disparities between our ontology and the one we found in the web mainly come from cultural considerations. For example, we do not found in the English ontology concepts like *terrace*.

Our domain ontology has been implemented under Protégé[5] and actually contains 239 concepts (from which we have 14 concepts directly related to the superclass owl:think), 36 object properties and 703 labels (646 labels for concepts and 57 labels for properties). The left part of figure 3 shows an extract of our restaurant domain ontology.

4.3 Opinion Lexicon

Our lexicon contains a list of opinion terms where each lexical entry is of the form:

[POS, opinion category, polarity, strength]

where *POS* is the part of speech tagging of the term, *opinion category* can be a judgment, a sentiment or an advice (see (Asher et al, 2009) for a detailed description of these categories), *polarity* and *strength* corresponds respectively to the opinion orientation (positive, negative and neutral) and the opinion strength (a score between 0 and 2). For example, we have the following entry for the term *good: [Adj, judgment, +, 1]*.

The lexicon actually contains 222 adjectives, 152 nouns, 157 verbs. It is automatically built following the algorithm described in (Chardon, 2010). We then add manually to this lexicon 98 adverbs and 15 expressions of negation.

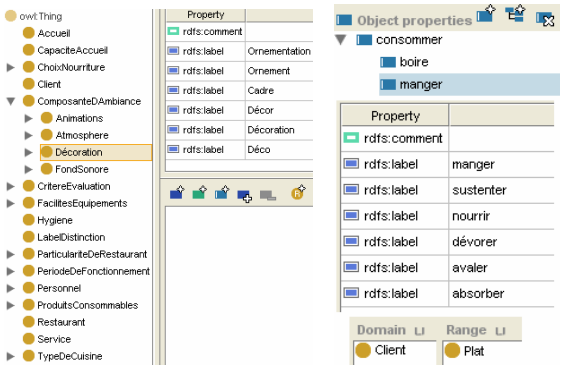

Figure 3. Extract of the restaurant domain ontology : Left - hierarchy of concepts and labels of "decoration" concept. Right – information about a particular object property.

4.4 Experiments

We conduct three types of experiment: the evaluation of the extraction of elementary opinion units (cf. section 3.1), the evaluation of the features extraction step (cf. section 3.2) and finally, the evaluation of the link between the retrieved opinion expressions and the retrieved object features (cf. section 3.3).

These experiments are carried out using GATE[6] toolkit. To evaluate our system, we create a gold standard by manually annotate in the corpus implicit and explicit elementary opinion units, implicit and explicit object features as well as for each opinion expression its associated feature.

Evaluation of the EOU extraction step.

The table below shows our results. Our system misses some EOU for two main reasons. The first one is due to missed opinion words in the lexicon and to implicit opinion expressions, such as *breathtaking*, since our extraction rules do not manage these cases (note that implicit opinion detection is still an open research problem in opinion mining).

[3] http://gaia.fdi.ucm.es/ontologies/restaurant.owl
[4] http://www.kelrestaurant.com/dept/31/ and http://www.resto.fr/default.cfm
[5] http://protege.stanford.edu/
[6] http://gate.ac.uk/

The second reason is the errors that come from the syntactic parser mainly because of typos and dependency link errors. Concerning precision, false positives are mainly due to some opinion words that are in our lexicon but they do not express opinions in the restaurant domain. In addition, some of our extraction rules, especially those that extract expression of recommendations, do not perform very well which imply a loss of precision.

Precision	**0,7486**
Recall	**0,8535**
F-measure	**0,7976**

Table 1. Evaluation of EOU extraction

Evaluation of the features extraction step.
Since the corpus is in the restaurant domain, the precision of this task is very good because most of the extracted features are relevant. However, recall is not as good as precision because the set of ontology labels do not totally cover the terms of the corpus. Another limitation of our system is that we do not take into account the cases where a term can be a linguistic realization of many concepts (ex. café can be a drink or a place to drink).
Figure 4 shows an example of the result we obtain for this step.

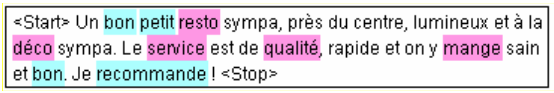

Figure 4. Result of EOU (blue) and ontological term (pink) extraction

Evaluation of the link between EOU and features.
The figure below shows our result on a sample. In this example, the system is able to extract opinion expressions which do not contain words present in the lexicon. It is the case with "*sympa (nice)*" which has been correctly associated to "*resto (restaurant)*" and "*deco (interior design)*" even if the word *nice* was not in the lexicon.
In order to evaluate the added value of using an ontology to feature based opinion mining, we compare our system to the well known approaches of Hu and Liu and Popescu and Etzioni (cf. section 2.1) that do not use any knowledge representation. We have also compared our approach to those that use taxonomies of concepts by deleting the properties of our domain ontology. The results are shown in table 2.

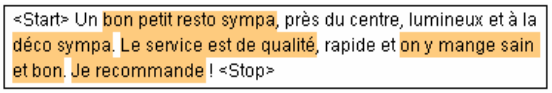

Figure 5. Result of linking EOU to extracted features

	Precision	Recall	F-measure
Our system	**0,7692**	**0,7733**	**0,7712**
Hu and Liu	0,6737	0,7653	0,7166
Popescu and al	0,7328	0,7387	0,7357
Taxonomy	0,7717	0,7573	0,7644

Table 2. Evaluation of our system and its comparison to existing approaches

In the Hu and Liu approach, features are nominal groups. We first extract all frequent features from our corpus that appear in more than 1% of the sentences. Then we extract EOU from those sentences (note that contrary to Hu and Liu, we do not extract only adjectives, but also nouns, verbs and adverbs). Non frequent features are finally removed as described in (Hu and Liu, 2004). In order to improve the extraction of relevant features, we extract features that have a good point mutual information value with the word *restaurant*, as described in (Popescu and Etzioni, 2005). The precision of our system is better compared to the approach of Hu and Liu that extracts too many irrelevant features (such as *any doubt, whole word*). Our system is also better compared to the PMI approach even if it performs better than Hu and Liu's approach. Recall is also better because our system can extract implicit features such as *well eating, lot of noise,* thanks to the use of ontology properties.
Finally, when using only taxonomy of concepts instead of the ontology, we observe that the F-measure is slightly better because actually fea-

tures related to object properties represent only 1,6% of feature cases in our corpus. Using, the ontology, our approach is able to extract from sentences like *"we eat good and healthy*" the couples (eat, good) and (eat, healthy) and then to link the opinion expressions to the concept *dish* whereas when using only the taxonomy, these opinion expressions are related to any feature.

5 Conclusion and prospects

5.1 Contribution of our system

Our method is promising because the use of the ontology allows to improve the feature extraction and the association between an opinion expressions and object features. On the one hand, the ontology is useful thanks to its concept list which brings a lot of semantic data in the system. Using concept labels the ontology allows to recognize terms which refer to the same concepts and brings some hierarchy between these concepts. On the other hand, the ontology is useful thanks to its list of properties between concepts which allows recognizing some opinions expressed about implicit features.

5.2 Prospects

Opinion lexicon improvement.
The opinion extraction we achieved is naive because we use a simple opinion word lexicon which is not perfectly adapted to the domain. To improve this part of the treatment, it would be interesting to use opinion ontology. As illustrated in section 2.2, constructing a domain ontology for the purpose of opinion mining poses several interesting questions in term of knowledge representation, such as: what are the frontiers between knowledge, where concepts are domain dependent, and opinion, where expressions can be at the same time dependent (the term *long* can be positive for a *battery life* but negative if it refers to a the *service* of a restaurant) and independent (the term *good* is positive) from a domain. Our intuition is that the two levels have to be separated as possible.

Natural Language processing (NLP) rules improvement.
Our system is limited by some current NLP problems. For example, the system does not treat the anaphora. For example, in the sentence "*Pizzas are great. They are tasty, original and generous*", it does not recognize that the three last adjectives refer to "*pizzas*". There is also the problem of conditional proposition. For example, in the sentence "*affordable prices if you have a fat wallet*", the system is not able to determine that "*affordable prices*" is subject to a condition.

Ontology and lexicon enrichment.
Thanks to the ability to link opinion expression and ontological term extractions, our system is able to extract some missing opinion words and labels of the ontology. We think it could be interesting to implement a module which allows the user to easily enrich opinion word lexicon and ontology. Furthermore, it will be interesting to evaluate the benefit of this method in both opinion mining and ontological domains.

Towards a discourse based summary.
The last step of the system is to produce a summary of the review that presents to the user all the opinion expressions associated to the main topic and all its features. This summary does not pretend to aggregate opinions for each feature or for the global topic. Instead, the aim is to organize the opinions of several reviews about one restaurant in order to allow the user to choose what feature is important or not for him.
In addition to this kind of summarization, we want to investigate how the domain ontology can be used to guide the process of identifying the most relevant discourse relations between elementary discourse units (EDU). Actually, the automatic identification of discourse relations that hold between EDUs is still an open research problem. Our idea is that there is continuation relation between EDU that contain terms that refer to concepts which are at the same level of the ontology hierarchy, and there is an elaboration relation when EDU contains more specific concepts than those of the previous clause.

References

Afantenos Stergos, Denis Pascal, Muller Philippe, Danlos Laurence. *Learning Recursive Segments for Discourse Parsing*. LREC 2010

Asher, Nicholas, Farah Benamara, and Yvette Y. Mathieu. 2009. *Appraisal of Opinion Expressions in Discourse*. Lingvisticæ Investigationes, John Benjamins Publishing Company, Amsterdam, Vol. 32:2.

Asher Nicholas and Lascarides Alex. *Logics of Conversation*. Cambridge University Press, 2003

BlairGoldensohn, Sasha, Kerry Hannan, Ryan McDonald, Tyler Neylon, George A. Reis, and Jeff Reynar. 2008. *Building a Sentiment Summarizer for Local Service Reviews*. WWW2008 Workshop : Natural Language Processing Challenges in the Information Explosion Era (NLPIX 2008).

Carenini, Giuseppe, Raymond T. Ng, and Ed Zwart. 2005. *Extracting Knowledge from Evaluative Text*. In Proceedings of the 3rd international conference on Knowledge captur.

Chardon Baptiste. *Catégorisation automatique d'adjectifs d'opinion à partir d'une ressource linguistique générique.* In proceedings of RECITAL 2010, Montreal, Canada

Pimwadee Chaovalit, Lina Zhou: Movie Review Mining: a Comparison between Supervised and Unsupervised Classification Approaches. HICSS 2005

Cheng, Xiwen, and Feiyu Xu. 2008. *Fine-grained Opinion Topic and Polarity Identification*. In Proceedings of the Sixth International Language Resources and Evaluation (LREC' 08), Marrakech, Morocco.

Feiguina, Olga. 2006. *Résumé automatique des commentaires de Consommateurs*. Mémoire présenté à la Faculté des études supérieures en vue de l'obtention du grade de M.Sc. en informatique, Département d'informatique et de recherche opérationnelle, Université de Montréal.

Gamon, Michael, Anthony Aue, Simon Corston-Oliver, and Eric Ringger. 2005. *Pulse: Mining Customer Opinions from Free Text*. In Proceedings of International symposium on intelligent data analysis N°6, Madrid.

Hu, Minqing, and Bing Liu. 2004. *Mining and Summarizing Customer Reviews.* In Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining.

Kim, Soo-Min, and Eduard Hovy. 2006. *Extracting Opinions, Opinion Holders, and Topics Expressed in Online News Media Text*. In Proceedings of ACL/COLING Workshop on Sentiment and Subjectivity in Text, Sydney, Australia.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. *Thumbs up? Sentiment Classification using Machine Learning Techniques*. Proceedings of EMNLP 2002.

Popescu, Ana-Maria, and Oren Etzioni. 2005. *Extracting Product Features and Opinions from Reviews.* In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing.

Read, Jonathon, David Hope, and John Carroll. 2007. *Annotating Expressions of Appraisal in English*. The Linguistic Annotation Workshop, ACL 2007.

Strapparava, Carlo, and Alessandro Valitutti. 2004. *WordNet-Affect: an Affective Extension of WordNet*. Proceedings of LREC 04.

Turney, Peter D. 2002. *Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews*. Proceedings of 2006 International Conference on Intelligent User Interfaces (IUI06).

Turney, Peter D., and Michael L. Littman. 2002. *Unsupervised Learning of Semantic Orientation from a Hundred-Billion-Word Corpus*. National Research Council, Institute for Information Technology, Technical Report ERB-1094. (NRC #44929)

Wiebe, Janyce, Theresa Wilson, and Claire Cardie. 2005. *Annotating Expressions of Opinions and Emotions in Language*. Language Resources and Evaluation 1(2).

Zhao, Lili, and Chunping Li. 2009. *Ontology Based Opinion Mining for Movie Reviews*. In Proceedings of the 3rd International Conference on Knowledge Science, Engineering and Management.

Author Index