Formal Description of Resources for Ontology-based Semantic Annotation

Yue Ma, Adeline Nazarenko, Laurent Audibert

Formal Description of Resources for Ontology-based Semantic Annotation

Yue Ma, Adeline Nazarenko, Laurent Audibert

Laboratoire d'Informatique de l'Université Paris-Nord (LIPN) - UMR 7030
Université Paris 13 - CNRS, France
givenname.name@lipn.univ-paris13.fr

Abstract

Ontology-based semantic annotation aims at putting fragments of a text in correspondence with proper elements of an ontology such that the formal semantics encoded by the ontology can be exploited to represent text interpretation. In this paper, we formalize a resource for this goal. The main difficulty in achieving good semantic annotations consists in identifying fragments to be annotated and labels to be associated with them. To this end, our approach takes advantage of standard web ontology languages as well as rich linguistic annotation platforms. This in turn is concerned with how to formalize the combination of the ontological and linguistical information, which is a topical issue that has got an increasing discussion recently. Different from existing formalizations, our purpose is to extend ontologies by semantic annotation rules whose complexity increases along two dimensions: the linguistic complexity and the rule syntactic complexity. This solution allows reusing best NLP tools for the production of various levels of linguistic annotations. It also has the merit to distinguish clearly the process of linguistic analysis and the ontological interpretation.

1. Introduction

The study of linguistic annotation on texts has a long tradition in Natural Language Processing. Several linguistic annotation platforms have been developed (Cunningham, 2002; Hamon et al., 2007), and a linguistic annotation standard has been proposed (Ide and Romary, 2004). Unlike linguistic annotation, semantic annotation is defined, in this paper, as the process of fixing the interpretation of a document by associating to it a formal and explicit semantic representation, which can be automatically handled. Among other possibilities, the semantic representation is expressed here with respect to an ontology.

Through this *ontology-based semantic annotation*, fragments of texts can be linked to elements of a domain ontology. This enables the automatic exploitation of text content. Applications can take advantage of ontology reasoning to remedy the information missing in texts or to improve text processing results. The annotation can also help the process of ontology modeling by referring to textual resources.

In theory, an ontology is defined as a "formal, explicit specification of a shared conceptualization" (Gruber, 1993). In other words, an ontology provides a set of shared concepts within a domain and the relationships between those concepts. It is used to reason about the underlying properties of that domain. OWL is an ontology language for making ontological statements which intends to be used over the World Wide Web. It has been standardized by W3C[1] and become one of the main techniques in the field of semantic web. OWL has the formal semantics defined by description logics (Baader et al., 2007). The use of OWL ontology can benefit from the state-of-the-art ontology reasoning tools, for instance to automatically infer implicit information underlying the domain knowledge.

Ontology-based semantic annotation takes texts as input on the one hand, and an ontology on the other hand. A given text will have different interpretations if different ontologies are considered. No matter which ontology is referred to, making semantic annotation on texts cannot avoid analyzing linguistic features of texts. One way to achieve an ontology-based semantic annotation system is to design a system from scratch which contains both linguistic analysis and the annotation with respect to the ontology. Instead of such an all-in-one paradigm, our semantic annotation model aims to allow implementations to be planted on the top of existing linguistic tools. That is, the system is decomposed into two separate layers, i.e. first the linguistic analysis and then the linkage to ontology. Through such a modularization, any state-of-the-art linguistic annotation tools can be reused in our semantic annotation system.

As we have seen, the ontology-based semantic annotation exploits two different resources: the linguistic and the ontological ones. Formalizing the combination of ontological and linguistical information is a topical issue that has got an increasing discussion recently (Buitelaar et al., 2006; Buitelaar et al., 2009; Cimiano et al., 2007; Montiel-Ponsoda et al., 2007; Reymonet et al., 2007). These studies aim to define either an ontology-based lexicon, a set of linguistic metadata to be used in the ontology, or a tight combination of linguistic information with the ontology. We adopt a different approach based on a loose coupling model that allows for the modularized ontology-based semantic annotation approach presented above. To this end, a unified resource representation, named *annotation ontology*, is proposed in this paper. It enables a clear distinction between the linguistic and ontology-based annotation processes, which can be summarized as follows:

- *Annotation ontology* is proposed as a formalism to encode resources for ontology-based semantic annotation. It extends the OWL ontology representation with extra *annotation rules*.

- *Annotation rules* are in the form of $AnnoCon \leftarrow AnnoPre$, where $AnnoCon$ and $AnnoPre$ are called the annotation conclusion and precondition, respectively. Annotation conclusions are ontological elements: concepts, roles, instances of concepts/roles, or axioms. Annotation preconditions can be of various

[1] http://www.w3.org

forms depending on their linguistic and syntactic complexity.

The annotation ontology is a resource formalism particularly specified for ontology-based semantic annotation. It provides rich annotation rules which are associated to an ontology and enable semantic annotations of various granularities: from the simple string-based annotation matching, to the disambiguated concepts/roles annotations, and to the annotation of newly discovered instances.

As a knowledge representation model, our work is similar to (Embley and Zitzelberger, 2010). But the latter does not focus on the semantic annotation of texts and do not take wide linguistic features into account. Compared to the existing related work on text semantic annotators (Dill et al., 2003; Kiryakov et al., 2004; Kokkinakis, 2008; Popov et al., 2003), this resource provides us with a broader sense of semantic annotation, which includes concepts and roles annotations from domain ontologies besides annotating instances of concepts/roles (aka. ontology population). It also has the merit to allow reusing best NLP tools.

This paper is structured as follows. The formalization of an annotation ontology is given in Section 2. Some concrete examples are used to illustrate the form of annotation rules in Section 3. The two dimensions existing in annotation rules are studied in Section 4. Related work is discussed in details in Section 5. Finally, we conclude this paper and present the future work in Section 6.

This work is illustrated with the corpus from AAdvantage Terms and Conditions (AAirline, 2009), document of American Airlines (AA), which explains mileage policy to customers.

2. Annotation Ontology: Extending ontologies with annotation rules

It has been discussed that simple labels on ontologies are not rich enough for deep level semantic annotation (Cimiano et al., 2007; Buitelaar et al., 2006; Montiel-Ponsoda et al., 2007), and we argue that the tight coupling of linguistic information in the ontological form (Buitelaar et al., 2006; Buitelaar et al., 2009; Cimiano et al., 2007) is not flexible enough to take advantage of the diversity of potential pre-existing linguistic annotations. Therefore, we need a more expressive but nevertheless flexible formalism to describe resources for semantic annotation.

First, we need to distinguish different types of annotations according to the nature of ontological elements which are used as annotations. Figure 1 illustrates this idea by an example:

1. Some words or expressions refer to ontological individuals. They are traditionally referred to as "named entities', such as "X Airline" which refers to a specific airline company, even if other types of noun phrases may also refer to individual entities such as "the minimum mileage guarantee", which is the name of a special policy. Those named entities must be related to the corresponding individuals or instances in the ontology.

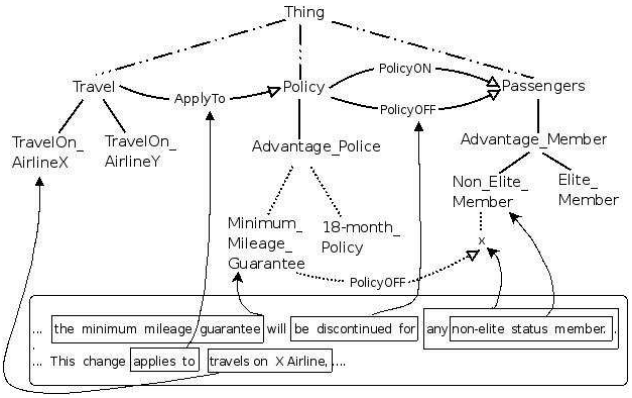

Figure 1: Ontology-based Semantic Annotation on texts "...the minimum mileage guarantee will be discontinued for any non-elite status member....This change applies to travels on X Airline,...".

2. Some words or expressions denote ontological concepts (e.g. "non-elite status member" or "travel on X Airlines"). They are usually referred as elements of the specialized domain vocabulary or domain terminology.

3. Similarly, terms may also denote conceptual roles if the underlying notions have been encoded as roles rather than as concepts in the ontology (e.g. "applies to", "be discontinued for" or "booking").

4. Some textual segments state relationships between individuals (e. g. "the minimum mileage guarantee will be discontinued for any non-elite status member").

5. Finally, some textual segments may express ontological axioms (e.g. in the sentence "Y is one of the world's largest global airline alliances", which can be interpreted as expressing a subsumption relation between the concepts "Y" and "Airline_Alliances").

These five types of annotations (individuals, concepts, conceptual roles, instantiated relations and axioms) are generally not considered all together. Some annotation processes focus on the population of ontologies (and thus on individuals and instantiated relations) whereas others rather consider the conceptual information or axiom discovery. In this paper, we consider all possibilities and propose a formalism, named *annotation ontology*, for encoding the resources required for such a deep semantic annotation. The annotation ontology is an ontology extended by annotation rules which is formally defined as follows.

Definition 1 *Suppose* $O = \langle \mathbf{C}, \mathbf{R}, \mathbf{I}, \mathbf{RI}, \mathbf{A} \rangle$ *is an ontology consisting of sets of concepts (*$\mathbf{C}$*), roles (*$\mathbf{R}$*), instances of concepts (*$\mathbf{I}$*), pairs of instances related with roles (*$\mathbf{RI}$*) and axioms (*$\mathbf{A}$*). Let* $\mathcal{R} = \langle \mathcal{R}_{\mathbf{C}}, \mathcal{R}_{\mathbf{R}}, \mathcal{R}_{\mathbf{I}}, \mathcal{R}_{\mathbf{RI}}, \mathcal{R}_{\mathbf{A}} \rangle$ *be a set of* annotation rules *which enable to annotate fragments of texts with concepts (*$\mathcal{R}_{\mathbf{C}}$*), roles (*$\mathcal{R}_{\mathbf{R}}$*), instances (*$\mathcal{R}_{\mathbf{I}}$*), relations between instances (*$\mathcal{R}_{\mathbf{RI}}$*) or axioms (*$\mathcal{R}_{\mathbf{A}}$*). Each annotation rule is in the form of* $AnnoCon \leftarrow AnnoPre$

where the conclusion is any element of O and the precondition identifies the text fragments that should be annotated by it. The annotation ontology*, written $O^{\mathcal{R}}$, is defined by extending O as follows:*

- *Each concept C of $\mathbf{C}$ is associated with a pair of sets of rules (ρ_c, ρ_i), where $\rho_c \in \mathcal{R}_{\mathbf{C}}$ and $\rho_i \in \mathcal{R}_{\mathbf{I}}$;*
- *Each role R of $\mathbf{R}$ is associated with a pair of sets of rules (ρ_r, ρ_{ri}), where $\rho_r \in \mathcal{R}_{\mathbf{R}}$ and $\rho_{ri} \in \mathcal{R}_{\mathbf{RI}}$;*
- *Each axiom a of $\mathbf{A}$ is associated with a set of rules $\rho_a \in \mathcal{R}_A$.*

Note that, for concepts and roles, there exist two sorts of annotation rules associated to them. The *conceptual rules* identify the occurrences or mentions of concepts (i.e. the annotation rule set $\mathcal{R}_{\mathbf{C}}$) and roles (i.e. the annotation rule set $\mathcal{R}_{\mathbf{R}}$) in a text. The *populating rules* identify the fragments of texts to be annotated and linked to (possibly new) instances of these concepts (i.e. the annotation rule set $\mathcal{R}_{\mathbf{I}}$) or roles (i.e. the annotation rule set $\mathcal{R}_{\mathbf{RI}}$).
The application of annotation rules on a corpus is an operation which returns a set of segments of the corpus to be annotated by ontological elements, thus enabling automatic semantic annotation.

3. Examples of Annotation Rule

Since our formalization allows for reusing existing linguistic resources, we assume that texts have been preprocessed beforehand by linguistic annotation tools, such as named entity recognizer, morpho-syntactic tagger, syntactic parser, etc. In the following examples, LEMMA, NAME_ENTITY, TERM, POS are used to represent the results of a lemmatizer, a named entity recognizer, a term tagger, and a part-of-speech tagger, respectively.
A fragment of an ontology on airline services is presented on Figure 2. We focus on three concepts which are descendants of the top concept "Thing": "Airline_Participant" (airline companies that participate the AAdvantage services), "AAdvantage_Member" (travelers who benefit from the AAdvantage services), and "Service". Moreover, as the set of registered customers of AAdvantage services, "AAdvantage_Member" can get "AAdvantage_Awards". Note that "Airline_Participant" and "AAdvantage_Member" are disjoint since their ancestors, company and people respectively, are essentially distinct from each other. Making a correct distinction between them is important for ontology-based semantic annotation. This will enable travelers, for example, to find policy entries which concern them rather than companies among a large number of evolving AAdvantage service documents.
Some linguistic annotators, such as named entity and term taggers, can assign conceptual categories to fragments of texts automatically. Here we take a specific term extractor YaTeA for example (Hamon and Aubin, 2006), which recognised 25 occurrences of "participant" and 81 occurrences of "member". A manual analysis showed that the annotation of these occurrences is not straightforward: six occurrences of "airline participant" and the ten occurrences of "AAdvantage member" have respectively the

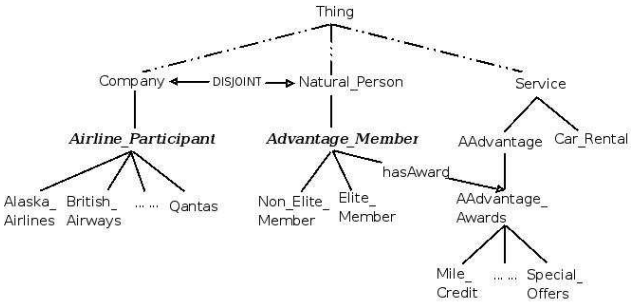

Figure 2: Sub-Ontology of Airline Services.

meaning of the concepts "Airline_Participant" and "AAdvantage_Member". However, 5 occurrences of "participant" actually refer to "AAdvantage_Member" and 2 occurrences of "member" means "Airline_Participant". Furthermore, several other fragments referring to either "Airline_Participant" or "AAdvantage_Member" in the corpus have not been discovered by YaTeA. Although not perfect, a term extractor like YaTeA does provide some useful information which can be reused for semantic annotation. This also confirms our assumption that semantic annotation should be grounded on existing linguistic resources such as linguistic annotation platforms.
As explained above, an annotation ontology consists of an ontology (e.g. Fig. 2) and a set of annotation rules (e.g. Tables 1 and 2) associated with it. The precondition of an annotation rule might itself be composed by two parts: the fragment description (noted in brackets which are not enclosed by brackets) aims at identifying the text fragments that are candidates for annotation as well as the optional contextual conditions (noted in angle brackets) which can filter out erroneous occurrences of ambiguous candidate fragments. The annotation rules presented here are written in an informal pseudo language for better comprehension but, since we do not commit to any specific format, the rules can refer to any linguistic annotation standard (Ide and Romary, 2004).
Let us consider the annotation rules in Table 1 for example. The annotation rules P1—P5 can trigger the annotations of concept $Airline\_Participant$ in the following sentences (the words in italic are the precise locations of the annotations):

S1: "AAdvantage mileage accrual eligibility on *airline participant* routes is subject to change without notice".

S2: "AAdvantage award restrictions may be announced by American Airlines or *AAdvantage participants* at any time without notice".

S3: "*AAdvantage participant* airlines and/or American Airlines codeshare flights...".

S4: "... please provide your AAdvantage number when you make your travel reservations or use the services of our *participants*".

S5: "*AAdavantage participant* is responsible for its awards only and not for the awards of other participating companies".

	Annotation Rule
P1	$Airline\_Participant \leftarrow$ [TERM = *airline participant*]
P2	$Airline\_Participant \leftarrow \langle$[LEMMA = *airline*][STRING = *or* $\vee$ *and*]$\rangle$ [TERM = *AAdavantage participant*]
P3	$Airline\_Participant \leftarrow$ [LEMMA = *participant*] $\langle$LEMMA = *airline*$\rangle$
P4	$Airline\_Participant \leftarrow \langle$[LEMMA = *service*][STRING = of]$^{\{0,3\}}\rangle$ [LEMMA = *participant*]
P5	$Airline\_Participant \leftarrow$ [LEMMA = *participant*] $\langle$[POS! = SENT]$^{\{0,20\}}$[LEMMA = *participate*][LEMMA = (*company* $\vee$ *carrier*)]$\rangle$
P3'	$Airline\_Participant \leftarrow$ [LEMMA = *member*] $\langle$LEMMA = *airline*$\rangle$

Table 1: Annotation rules for the concept $Airline\_Participant$

	Annotation Rule
P1'	$AAdvantage\_Member \leftarrow$ [TERM = *AAdvantage member*]
P6	$AAdvantage\_Member \leftarrow$ [TERM = *AAdavantage participant*] $\langle$[POS! = $SENT$]$^{\{0,20\}}$[LEMMA = *award*] [STRING = *for*][ ]$^{\{0,3\}}$[LEMMA = *participant*]$\rangle$
P7	$AAdvantage\_Member \leftarrow$ [TERM = *participant*] $\langle$SUJPAS(LEMMA = (*affiliation* $\vee$ *miles*), LEMMA = *earn*)$\rangle$

Table 2: Annotation rules for the concept $AAdvantage\_Member$

The first annotation rule (P1) says that the exact matching of the term "airline participant" is enough for a fragment to be annotated as an occurrence of the concept $Airline\_Participant$. P1' in Table 2 is similar to P1 but other annotation rules combine context restrictions with the fragment description. Note that, although "AAdvantage participant" refers to the concept $Airline\_Participant$ in the second, third, and fifth sentences above, the sole presence of the term "AAdvantage participant" cannot become a conceptual rule for the concept $Airline\_Participant$. Indeed, a counterexample is the following sentence that matches "AAdvantage participant" but refers to the concept $AAdvantage\_Member$, which can be discovered by the annotation rule P6 from Table 2.

S6: "If an *AAdvantage participant* agreement changes or terminates, you may find that AAdvantage awards for that participant are no longer available".

Rule P7 in Table 2, whose details are explained in the next section, is another annotation rule which enables the sentence below to be annotated with the concept $AAdvantage\_Member$,:

S7: "Your summary includes flight and *participant* mileage earned, along with AAdvantage program information and special promotions".

Another point worth noticing is that P3 and P3' have the same precondition except for the fragment descriptions ([LEMMA = *participant*] vs. [LEMMA = *member*]). Actually, P3' is constructed via replacing "participant" by "member" in P3. Obviously, P3' can make a semantic annotation w.r.t. to concept $Airline\_Participant$ instead of $AAdvantage\_Member$ in the following sentence.

S3': "In addition, in some instances the minimum mileage amount earned may be less than 500 miles for travel on oneworld *member* airlines and AAdvantage participanting airlines".

We can see, from Tables 1 and 2, that the annotation rules do not always have the same complexity.

4. Annotation Rule Complexity

While the conclusions of annotation rules are well defined by ontology language, preconditions can either be written by users or computed from training corpora. In this section, we discuss the two dimensions of complexity existing in annotation preconditions: *linguistic complexity* and *rule syntactic complexity*. Note that the "rule syntactic complexity" refers to the complexity of the rule language, and not to the complexity of the linguistic elements on which the rule applies, which is referred as "linguistic complexity". A clear distinction among different descriptive complexities in an annotation rule has the following benefits:

- It gives a flexible way to reuse existing linguistic resources (e.g. the linguistic annotations produced by annotation platforms).
- Since a simpler complexity means a more efficient computability but less expressivity, for annotation rule construction algorithms, the balance of efficiency and expressivity can be achieved by choosing a proper level of complexity.

For terminological clarification, an undecomposable unit needed to be matched for the truth of the precondition is called an *item*.

4.1. Linguistic Complexity

We divide the linguistic features that are used in annotation preconditions into the following different levels (horizontal-axis of Figure 3): *plain strings, lemmas, named entities, terms, morpho-syntactic categories and features, and syntactic relations*. In the following, we indicate how they are used in annotation rules.

If a precondition item is a plain string, triggering such an annotation rule will require an exact match between the

segment and the specified string. For example, the annotation rule $Elite\_AAdvantage\_Member \leftarrow [\text{STRING} = \textit{elite advantage member}]$ will produce semantic annotations $Elite\_AAdvantage\_Member$ on texts which contain the string "elite advantage member". Obviously, this complexity is not enough to discover interesting semantic antnations in many real applications.

Unlike plain strings, other levels of complexity consider linguistic features into semantic annotation rules, which is also one of the main differences from (Ding et al., 2006; Embley and Zitzelberger, 2010).

Lemma is the canonical form of a word. In a real usage, a word is considered as a lemma combined with flexional affixes (mainly suffixes) that bear morphological features such as tense, person, case, and number for verbs. For instance, $[\text{LEMMA} = airline]$ used in the annotation rule P2 of Table 1 allows for the matching of textual fragments " Airlines". Similarly, the specification $[\text{LEMMA} = \textit{participate}\,][\text{LEMMA} = \textit{company} \vee \textit{carrier}\,]$ will accept the textual span "participating companies".

One level more complex than lemmas is the *named entity* level. Named entities refer to meaningful linguistic units of a text, such as car brand (e.g. VW, NISSAN), people name (e.g. Cyrille Bissette), Phone numbers, etc. The interesting point is that named entity recognition tools can help to check various forms of mentions of a same named entity. For example, M. Bissette, C. Bissette, and Cyrille Bissette are all tagged as $\text{NAMED\_ENTITY} = \textit{Cyrille Bissette}$. In addition, a named entity recognizer (e.g. Stanford NER (Finkel et al., 2005), LBJ NER (Ratinov and Roth, 2009)) can return the type of the named entity, which is marked by TYPE in this paper. For instance, $[\text{TYPE} = people]$ will match "Mr. Cyrille Bissette". In our example, "18-month" is a named entity for named entity recognizers, which refers to a special period of time. Then we can have $\textit{Policy}_i \leftarrow [(\text{NAME\_ENTITY} = \textit{18-month})(\text{LEMMA} = \textit{policy})]$ as a population rule for the concept "Policy". By this annotation rule, "18-month policy" is annotated as an instance of the concept "Policy", not relating to time any more. As most of the named entities talk about the instances, population rules for concepts and roles will often contain named entities as components.

A level higher than named entities, are the *terms*. A set of terms is about specific terminologies acknowledged by a large majority of experts in specific domains, and

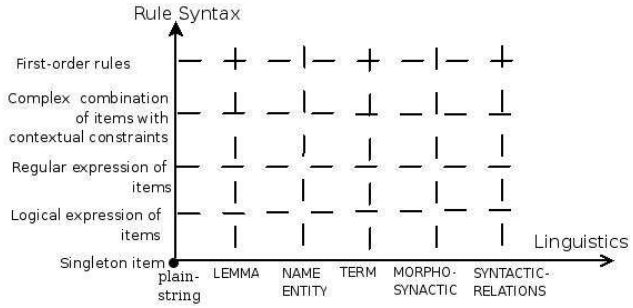

Figure 3: Complexity of Annotation Rule.

term extractors or annotators help to identify specific textual units as a terms which often contains several words. For example, an annotation precondition in the form of $[\text{TERM} = \textit{airline participant}]$ allows for annotating "participating airlines" despite its morphological variation and the permutation of words. Several term extractors could be used by our formalization, including YaTeA (Hamon and Aubin, 2006), TermExtractor (Velardi and Sclano, 2007), TermFinder[2], Acabit[3], etc.

Another linguistic complexity is related to morpho-syntactic features, usually part-of-speech (POS) of a word (Noun, Verb, etc.) or phrase (Noun Phrase, Verb Phrase, etc). Enabling morpho-syntactic features in an annotation rule is important. For example, the annotation rule P5 in Table 1 provides a way to say that it is only applicable to fragments of texts which contain a lemma "participant" followed by the lemma in the form of "participant company" or "participant carrier" in the same sentence (expressed by $[\text{POS!} = \text{SENT}]^{\{0,20\}}$) within a distance of 20 words, where SENT is a special POS feature meaning the end of a sentence.

The most complex linguistic item that we consider in annotation rules is the *syntactic relations*. A syntactic relation defines the role (or function) played by a word with respect to another one. Normally a syntactic relation can be represented by a triplet *Type(Head, Modifier)*, where *Type* is the relation type (e.g. subject of a verb, subject of a passive verb, etc) between the *Head* and the *Modifier*. As an example, consider the annotation rule P7 in Table 2 whose context contains a syntactic relation statement $\text{SUJPAS}(\text{LEMMA} = (\textit{affiliation} \vee \textit{mileage}), \textit{earn})$. It represents the "subject of a passive verb" relation, denoted by $\text{SUJPAS}(\cdot,\cdot)$, between the word whose lemma form is either *affiliation* or *mileage* and the verb "earn". It is by this annotation rule that the sentence S7 can be matched and annotated as an occurrence of concept *AAdvantage_member* instead of *Airline_paticipant*. An NLP annotation platform, such as GATE (Cunningham, 2002; Hamon et al., 2007), usually proposes a syntactic parsing in complement to other linguistic annotation layers.

It is not surprising to see that the linguistic complexity presented above corresponds to standard tasks for linguistic annotation platforms. The implementations of our annotation approach can rely on the best NLP tools. The annotation rules are used to distinguish and link the processes of linguistic analysis and the ontological interpretation.

4.2. Rule Syntactic Complexity

As shown previously, an annotation precondition can consist of two parts: a fragment description and optional contextual conditions. The rule syntactic complexity of annotation preconditions (vertical-axis of Figure 3) depends on whether contextual constraints are used and how complex the fragment description is. We distinguish propositional rules and first-order rules, the first category itself consisting of four subcategories of rules depending on the type of the fragment description which can be a singleton item, a

[2]http://labs.translated.net/terminology-extraction/

[3]http://cl.it.okayama-u.ac.jp/rsc/jacabit/index.html

logical expression of items or a regular expression for sequences of items, and on the presence of additional contextual constrains.

An annotation precondition where the fragment description consists of a single item has the simplest syntax, such as [TERM $= Policy$] or [STRING = *AAdvantage member*]. At this level, regular expressions of string, such as [STRING $\sim [Aa]irline\$$] which matches "Airline" and "airline" where$\sim$ denotes the matching with a regular expression, are allowed but only at the end of the string or line.

More complex is the logical expression of items (conjunctions, disjunctions, negations of items). For example, a set of synonyms can be expressed as [[STRING = *passenger*]$\vee$[LEMMA = *Traveler*]] which is the disjunction of a plain string item and a lemma item. Similarly, the logical expression $[(\text{LEMMA} \sim \ ^\wedge L) \wedge (\text{POS!} = N))]$ represents an item beginning with "L" but different from a noun. Note that each item in the logical expression can be of any level of linguistic complexity.

Above the logical expression of items is the regular expression of items which can describe a sequence of items. For instance, we can have [[STRING $= Mr.$][STRING $= *]^{\{0,3\}}$[STRING $\sim \ ^\wedge[A-Z][a-z]*$]] for annotating fragments beginning with "$Mr.$" followed by at most three words and a word with a capital initial, as the precondition of a populating rule of the concept *Person*. More regular expressions can be found in Table 1 and Table 2, such as the context of annotation rules P2, P5, P6, and P7.

A further complexity comes from the introduction of contextual constraints on all of the previous cases, such as in the annotation rules P2–P7 in Tables 1 and 2, each of which containing a left or/and a right context condition. Another example is an annotation rule for the role *Apply_To*: $Apply\_To \leftarrow \langle \text{TERM} = policy\rangle[\text{LEMMA} = apply]\langle \text{STRING} = to\rangle$ which has the contextual constraint of "TERM $= Policy''$ and "STRING $= to''$ surrounding the main item "[LEMMA $= apply]''$.

More complex annotation rules can even take the form of first-order rules with variables, such as the role populating rule which interests the LLLchallenge[4]: *gene_interaction*$(X, Z) \leftarrow$ SUBJECT(X, Y), X.TYPE $=$ *protein*, Y.POS $= V$, Y.TYPE $=$ *interaction_action*, OBJD(Z, Y), Z.TYPE $=$ *gene-expression*, where X.TYPE $=$ *protein* indicates that the type of X returned by a named entity tagger must be *protein*, Y.POS $= V$ requires Y to be a verb, and OBJD(Z, Y) means that Z is the object of Y.

As shown in Figure 3, linguistic and syntax complexities can mix together in preconditions of annotation rules, which is the case of annotation rules in Tables 1 and 2. The origin of Figure 3 is the simplest case (single string). But obviously its expressivity is poor and it will miss many interesting semantic annotations. More useful annotation rules should allow disjunctions of plain strings and lemmas that can be obtained from thesauri. It will be better if other linguistic annotations of even higher levels available, such as named entity or term annotations. But this is not enough for annotation rules which conclude on a role. For example, "*Membership*" relation is ambiguous, in the same way as "*participant*" in our running example, because in the corpus both of them are used sometimes to refer relations between a person and a human organization, such as "*Mr. Thomas* is a member of X program", and sometimes between a company and an alliance of companies, such as "*X Airline is a member of Y alliance*". In order to solve such ambiguities, we need annotation rules with contextual condition to check if a person or a company is mentioned in the context, rather than a simple disjunction of items like "member of". For populating rules, regular expressions of items are helpful as under-specified fragment descriptions enable the identification of unknown instances through the recognition of new named entities.

5. Related Work

There actually exist several models to represent the combination of lexicons (or terminologies) and ontologies. In this section, we summarize them and conclude why we adopt a different model from theirs.

For the task of text-oriented semantic annotation, the established W3C models, including OWL (W3C-OWL, 2004), its development OWL 2 (W3C-OWL2, 2009), RDF and RDFS (W3C-RDFS, 2004), are weak. This is because these formalisms are quite restricted to allow for linguistic information. Indeed, they only provide a way to define labels for ontological elements using rdf:label property. But the range of rdf:label is *Literal*, which limits the expression of more complex linguistic information.

SKOS (W3C-SKOS, 2008), currently under development at W3C, aims at producing a data model for representing classification schemas to enable easy publication of controlled structured vocabularies for the Semantic Web. However, the association of linguistic information and ontologies is not considered in SKOS.

LMF (Francopoulo et al., 2007; LMF Working Group, 2008) is an ISO standard for Natural Language Processing (NLP) lexicons and Machine Readable Dictionaries (MRD). LMF arrived at a coherent UML model that represents lexicons in detail. Its extension named LexInfo (Buitelaar et al., 2009) is a metamodel for ontology-based lexicons which is based on two previously developed models, LingInfo (Buitelaar et al., 2006) and LexOnto (Cimiano et al., 2007). Both LMF and LexInfo metamodels have been formalized using OWL language[5]. They aim to capture in a declarative way (using an ontology) the relation between the way we talk about things and the way they are formalized in a given ontology. For example, LexInfo gives a sound and principled model to exchange lexicons across systems. This goal is different from ours. Technically speaking, our work is also different in the way we model the mapping between linguistic features and the conceptual ontology. Under LMF and LexInfo, the mapping is modeled by ontology axioms. For example, LMF has a general mechanism which allows to associate *LexicalEntry* objects with a *Sense* via *hasSense* property claimed in the

[4] http://genome.jouy.inra.fr/texte/LLLchallenge/

[5] They are downloadable from http://lexonto.ontoware.org/lmf and http://lexonto.ontoware.org/lexinfo, respectively.

LMF ontology, which is inherited by LexInfo. On the contrary, in our work, the linguistic features are associated to ontological elements as their annotations such that the linguistic information is not involved in ontology reasoning, which is not the case of LMF or LexInfo model. We claim that for the semantic annotation goal, our model provides more flexibility than LexInfo but is still expressive.

Another related formalization is the *extraction ontology*, provided in (Ding et al., 2006; Embley and Zitzelberger, 2010), which allows regular expressions of strings to be used as an external representation, and strings to be used as context keywords associated with ontological elements. However, this corresponds to two specific cases in our formalization in terms of rule complexity. Moreover, the only linguistic information considered in the extraction ontology is the lexicon and the synonyms, which are again more restricted than our annotation ontology.

In the context of multilingual issues, OMV (Ontology Metadata Vocabulary) (Hartmann et al., 2005) proposes standard metadata descriptors for ontologies. LexOMV (Montiel-Ponsoda et al., 2007) extends the OMV model by providing metadata which can describe the lexical level of ontological elements such that ontology users can know more information about the linguistic features of ontologies. Instead of multilinguality, our current work focuses on the mechanism for detecting semantic annotations as rich as possible. Therefore, multilingual features are not considered in our annotation ontology formalization, although it may be an interesting topic for future work.

6. Conclusion and Future Work

We have motivated and formally described a language resource, namely the extended ontology associated with annotation rules, for semantic annotation. According to the expressivity of OWL ontology language, we have explained that annotations can have five types. By several illustrative examples, it has been shown that existing state-of-the-art information extractors are not enough for making semantic annotations. Therefore, annotation rules have been introduced as a flexible resource specified for the aim of semantic annotation, which can largely reuse the existing information extractors and NLP tools. The two dimensions of complexity existing in annotation rules, i.e. the linguistic complexity and rule syntactic complexity, have been carefully defined in this paper.

Obviously, much work remains to be done to make the language resource proposed in this paper exploitable in practice. We are actually working on the creation of semantically annotated corpora as well as on tools for acquiring, operating and applying such a resource.

Acknowledgements

This work was realized as part of the Quaero Programme (funded by OSEO, French State agency for innovation) and of the FP7 231875 ONTORULE project. We are thankful to American Airline who is the owner of the corpus that is used as a working example within the ONTORULE project.

7. References

AAirline. 2009. AAdvantage terms and conditions. http://www.aa.com/i18n/AAdvantage/programDetails/termsAndConditions/termsAndConditions.jsp (Version March, 2009).

Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider, editors. 2007. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, second edition.

Paul Buitelaar, Michael Sintek, and Malte Kiesel. 2006. A multilingual/multimedia lexicon model for ontologies. In York Sure and John Domingue, editors, *Proceedings of 3rd European Semantic Web Conference (ESWC'06)*, volume 4011 of *Lecture Notes in Computer Science*, pages 502–513. Springer.

Paul Buitelaar, Philipp Cimiano, Peter Haase, and Michael Sintek. 2009. Towards linguistically grounded ontologies. In Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero Hyvönen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou, and Elena Paslaru Bontas Simperl, editors, *Proceedings of 6th European Semantic Web Conference (ESWC'09)*, volume 5554 of *Lecture Notes in Computer Science*, pages 111–125. Springer.

P. Cimiano, P. Haase, M. Herold, M. Mantel, and P. Buitelaar. 2007. Lexonto: A model for ontology lexicons for ontology-based nlp. In *Proceedings of the International OntoLex Workshop, held in conjunction with ISWC*, pages 1–12.

Hamish Cunningham. 2002. Gate, a general architecture for text engineering. In *Computers and the Humanities*, volume 36, pages 223–254. Springer.

Stephen Dill, Nadav Eiron, David Gibson, Daniel Gruhl, R. Guha, Anant Jhingran, Tapas Kanungo, Sridhar Rajagopalan, Andrew Tomkins, John A. Tomlin, and Jason Y. Zien. 2003. Semtag and seeker: bootstrapping the semantic web via automated semantic annotation. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 178–186. ACM.

Yihong Ding, David W. Embley, and Stephen W. Liddle. 2006. Automatic creation and simplified querying of semantic web content: An approach based on information-extraction ontologies. In Riichiro Mizoguchi, Zhongzhi Shi, and Fausto Giunchiglia, editors, *Proceedings of the first Asian Semantic Web Conference (ASWC'06)*, volume 4185 of *Lecture Notes in Computer Science*, pages 400–414. Springer.

David W. Embley and Andrew Zitzelberger. 2010. Theoretical foundations for enabling a web of knowledge. In Sebastian Link and Henri Prade, editors, *6th International Symposium Foundations of Information and Knowledge Systems (FoIKS'10)*, volume 5956 of *Lecture Notes in Computer Science*, pages 211–229. Springer.

Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of*

the Conference (ACL'05). The Association for Computer Linguistics.

Gil Francopoulo, Nuria Be, Monte George andNicoletta Calzolari, Monica Monachini, Mandy Pet, and Claudia Soria. 2007. Lexical markup framework: Iso standard for semantic information in nlp lexicons. In *Proc. of the Workshop of the GLDV Working Group on Lexicography at the Biennial Spring Conference of the GLDV*.

Thomas R. Gruber. 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, June.

Thierry Hamon and Sophie Aubin. 2006. Improving term extraction with terminological resources. In Tapio Salakoski, Filip Ginter, Sampo Pyysalo, and Tapio Pahikkala, editors, *Advances in Natural Language Processing 5th International Conference on NLP (FinTAL'06)*, pages 380–387. Springer.

Thierry Hamon, Julien Derivière, and Adeline Nazarenko. 2007. Ogmios: a scalable NLP platform for annotating large web document collections. In *Procedings of Corpus Linguistics 2007*, Birmingham, UK, July.

Jens Hartmann, York Sure, Peter Haase, Raúl Palma, and Mari del Carmen Suárez-Figueroa. 2005. OMV – Ontology Metadata Vocabulary. In Chris Welty, editor, *ISWC 2005 - In Ontology Patterns for the Semantic Web*, NOV.

Nancy Ide and Laurent Romary. 2004. International standard for a linguistic annotation framework. *Natural Language Engineering*, 10(3-4):211–225.

Atanas Kiryakov, Borislav Popov, Ivan Terziev, Dimitar Manov, and Damyan Ognyanoff. 2004. Semantic annotation, indexing, and retrieval. *J. Web Sem.*, 2(1):49–79.

Dimitrios Kokkinakis. 2008. A semantically annotated swedish medical corpus. In Nicoletta Calzolari et al., editors, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*. ELRA.

LMF Working Group. 2008. Language resource management — lexical markup framework (lmf), iso/tc37/sc 4 n453 (n330 rev.16). Technical report.

E. Montiel-Ponsoda, G. Aguado de Cea, M.C. Suarez-Figueroa, R. Palma, W. Peters, and A. Gomez-Perez. 2007. LexOMV: an OMV extension to capture multilinguality. In *Proceedings of OntoLex'07*, pages 118–127.

Borislav Popov, Atanas Kiryakov, Dimitar Manov, Angel Kirilov, Damyan Ognyanoff, and Miroslav Goranov. 2003. Towards semantic web information extraction. In *In Human Language Technologies Workshop*, Florida, USA, 20 October 2003. Proceedings of ISWC workshop.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL '09:)*, pages 147–155. Association for Computational Linguistics.

Axel Reymonet, Jérôme Thomas, and Nathalie Aussenac-Gilles. 2007. Modélisation de ressources termino-ontologiques en owl. In Francky Trichet, editor, *Proceedings of the 18th French Knowledge Engineering Conference (Actes d'IC 07)*, pages 169–181. Cepadues.

Paola Velardi and Francesco Sclano. 2007. Termextractor: a web application to learn the common terminology of interest groups and research communities. In *9th Conf. on Terminology and Artificial Intelligence (TIA'07)*.

W3C-OWL. 2004. Web ontology language reference. W3C Recommendation (http://www.w3.org/TR/owl-ref).

W3C-OWL2. 2009. Web ontology language reference. W3C Recommendation (http://www.w3.org/TR/owl2-profiles/).

W3C-RDFS. 2004. Rdf vocabulary description language 1.0: Rdf schema. W3C Recommendation (http://www.w3.org/TR/rdf-schema/).

W3C-SKOS. 2008. Simple knowledge organization system reference. W3C Working Draft (http://www.w3.org/TR/skos-reference/), 29 August.