On the Semantic Representation and Extraction of Complex Category Descriptors

André Freitas[1], Rafael Vieira[2], Edward Curry[1], Danilo Carvalho[3], João C. Pereira da Silva[2]

[1]Insight Centre for Data Analytics, National University of Ireland, Galway
[2]Computer Science Department, Federal University of Rio de Janeiro (UFRJ)
[3]PESC/COPPE, Federal University of Rio de Janeiro (UFRJ)

Abstract. Natural language descriptors used for categorizations are present from folksonomies to ontologies. While some descriptors are composed of simple expressions, other descriptors have complex compositional patterns (e.g. 'French Senators Of The Second Empire', 'Churches Destroyed In The Great Fire Of London And Not Rebuilt'). As conceptual models get more complex and decentralized, more content is transferred to unstructured natural language descriptors, increasing the terminological variation, reducing the conceptual integration and the structure level of the model. This work describes a formal representation for complex natural language category descriptors (NLCDs). In the representation, complex categories are decomposed into a graph of primitive concepts, supporting their interlinking and semantic interpretation. A category extractor is built and the quality of its extraction under the proposed representation model is evaluated.

1 Introduction

Ontologies, vocabularies, taxonomies and folksonomies provide structured descriptors for categories of objects and their relationships. While ontologies target a more centralised, consistent and structured representation of a domain, folksonomies allow a decentralised, less structured categorization. Both representation models have in common natural language descriptions associated with object categories. These *natural language category descriptors* (NLCDs) are a fundamental part of the communication of the meaning behind these artefacts. While some descriptors are composed of single words or simple expressions (e.g. 'Person', 'Country', 'Film'), other descriptors have more complex compositional patterns (e.g. 'French Senators Of The Second Empire', 'United Kingdom Parliamentary Constuituencies Represented By A Sitting Prime Minister').

NLCDs are used to describe categories, sets of objects and attributes. As the complexity of the domain of discourse and the decentralization of the content generation increases in contemporary data management environments, more effort is necessary for defining a consistent and structured conceptual model. As a consequence, as the scale of the domain of discourse increases, data representation strategies move from more structured conceptual models to less structured categorization systems (e.g. folksonomies), with an impact on the ability of users to analyse and query the data.

This shift from structured towards more unstructured conceptual models is reflected in the content and structure of NLCDs. As models get more complex and decentralized, more content is transferred to unstructured natural language descriptors, increasing the terminological variation, reducing the conceptual integration and the structure level of the model. In this scenario, the more formal conceptual model tools are substituted by complex NLCDs as an interface for domain description. From the perspective of information extraction and representation, NLCDs provide a much more tractable subset of natural language which can be used as an '*interface*' for the creation of structured domains. From the syntactic perspective, natural language category descriptors (NLCDs) are short and syntactically well-formed phrases. Differently from full sentences, NLCDs present simpler and more regular compositional patterns. By structuring NLCDs, we intend to support the creation of more structured resources with lower construction effort and in a more decentralized way, partially addressing the structure level/construction effort trade-off.

In this work we describe an extraction and a formal representation approach for complex NLCDs. In order to understand the structure of NLCDs we analyse Wikipedia categories (section 4). This analysis will support the construction of a representation (section 5) and an extraction (section 6) approach. In the representation, complex predicates are decomposed into a graph of primitive word senses supporting the alignment between different NLCDs. A NLCD extractor is built and the extraction quality is evaluated (section 7).

2 Motivational Scenario

Wikipedia is built from a large-scale decentralized data curation effort. It is estimated that more than 300,000 editors have edited Wikipedia more than 10 times[1], being one of the largest scale examples of decentralized data curation effort. In addition to the unstructured data content, Wikipedia pages contain structured/semi-structured data such as infoboxes and category links. Datasets such as DBpedia and YAGO are built automatically from the extraction of Wikipedia's structured data. For example, some of the classes of DBpedia and YAGO are derived from category links (e.g. RussianFemaleCosmonauts). This rich classification structure can benefit users by providing additional information in the datasets, supporting querying and data analysis. The information for answering the query '*Who are the Soviet Female Astronauts?*' can be answered by using this decentralized categorization system.

However, there is an intrinsic cost associated with using a decentralized categorical structure. Due to the large number of possible word compositions in the creation of the class, from the data consumption side the number of classes can grow unmanageable. The number of Wikipedia categories (YAGO/DBpedia

[1] http://en.wikipedia.org/wiki/Wikipedia:Wikipedians

classes) exceeds 280,000: at this scale, it is not feasible for data consumers to read all the classes which are available in order to build a structured query.

Ideally users should be abstracted away from the representation of categorical descriptors. However, categories can be expressed using different lexical expressions and abstraction levels and a user may express the same class using different terms (e.g. SovietAstronauts, MoscoviteWomen, CosmonautsFromTheUSSR, etc). In order to provide this flexible interpretation, this work proposes an extraction and representation model for natural language category descriptor (NLCDs). The relevance of the proposed approach lies in the fact that NLCDs are intrinsic elements in semi-structured and structured data and on their increasing importance as categorization artefacts in decentralised data management systems.

3 Related Work

Different works have focused on information and data extraction approaches applied in the context of semantic annotations and the Semantic Web. Most of these approaches have targeted the extraction of ontologies and datasets from structured data [1], from unstructured data [7] or the alignment of folksonomies to ontologies [3][4][6]. To the best of our knowledge no existing work have focused on the extraction and representation of complex natural language category descriptors.

YAGO [1] is a large-scale ontology which is automatically built from Wikipedia and WordNet. YAGO extracts facts from the infoboxes and the category system of Wikipedia, representing them in a data model which is based on reified RDF triples, expressing relations between facts and n-ary relations. YAGO builds a taxonomic structure from Wikipedia categories, aligning them to WordNet synsets.

In comparison to YAGO, this work provides a model focussed on extracting and representing complex predicates. Similarly to YAGO, the data model used to represent the complex predicates is based on RDF with reified triples. However this model is specialised to cope with particular representation demands from complex predicates (section 4). The alignment between YAGO and WordNet synsets is done [1] by taking the most frequent WordNet senses for the head term of the complex class. We argue that despite being a good strategy for high-level classes (e.g. Person, Place), it can't be generalized for the other terms of the class. This work uses a distributional semantics based Word Sense Disambiguation (WSD) strategy.

Specia & Motta [3] proposes an approach for making explicit the semantics behind the tags, by using a combination of shallow pre-processing strategies and statistical techniques, together with knowledge provided by ontologies available on the Semantic Web. The final result consists in the generation of clusters with related tags corresponding to concepts in ontologies. Cattuto et al. [4] proposed a systematic evaluation of similarity measures for folksonomies. Voss [6] concentrates on the description of an approach for the translation of folksonomies to Linked Data and SKOS. Comparatively, most of the previous works concentrate on the analysis and alignment of simple (non-complex) tags. Another difference is the proposal of a representation model which goes beyond a taxonomic structure.

4 The Structure of Natural Language Category Descriptors (NLCDs)

In order to understand the syntactic structure of NLCD descriptors, an analysis based on the complete set of Wikipedia category links was performed. The complete set contains 287,957 categories. The goal of this analysis is to derive a representation model which can express the relationships between the concepts of the classes following a Semantic Web compatible graph data model. The analysis process started with the manual analysis and categorization of a random sample of 10,000 categories, in order to derive a set of recurrent representation (features) present in the query. Table 1 shows the set of category features and instances of categories.

Features	**Category Examples**
Classes with verbs	United Kingdom Parliamentary Constituencies Represented By A Sitting Prime Minister, Local Government Districts Created By The Local Government Act 1858
Classes with temporal references	19th-century Presidents Of The United States, Tennis Players At The 1996 Summer Olympics
Classes with named entities	Olympic Gold Medalists For The United States, Populated Places In North Holland
Classes with conjuctions	Former Buildings And Structures Of The City Of London, Alumni Of The School Of Oriental And African Studies
Classes with disjuctions	Nobel Laureates In Physiology Or Medicine, Snow Or Ice Weather Phenomena, Converts To Christianity From Atheism Or Agnosticism
Classes with operators	Dutch Top 40 Number-one Singles, World No.1 Tennis Players, Ships Of The First Fleet, Cricketers Who Have Played For More Than One International Team

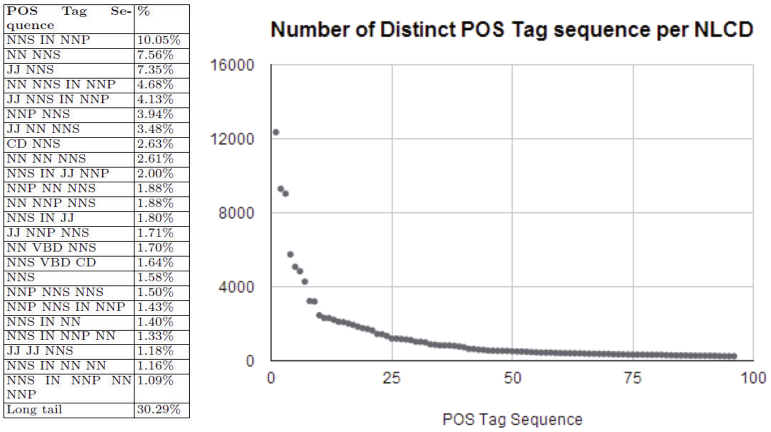
Table 1: Core feature set and examples of categories with different feature types.

The manual analysis showed an enumerable set of recurrent features in the NLCDs. After the determination of the core representation features, we automatically analysed the complete set of 287,957 NLCDs, according to the incidence of the features. Table 2 shows the distribution of features in the full category set. The typical NLCD consists of an entity described by two or more words with one or more specialization relations and it mainly consists of one or more nouns specialized by an adjective. There is, however, a significant variability in the combination of the features set present at the category collection.

The possible combination of features follows a long tail distribution which is expressed in the distribution of the sequence of POS Tags for the categories (Figure 1). A total of 96 distinct POS Tag sequences were found.

# of Features	Operators	Words	Proper Nouns	Nouns	Adjectives	Verbs
0	99.846%	0%	46.348%	1.461%	62.284%	81.808%
1	0.154%	15.818%	46.594%	40.173%	32.089%	17.373%
2		26.618%	6.794%	39.727%	5.078%	0.814%
3		24.507%	0.226%	14.572%	0.504%	0.004%
4		18.612%	0.036%	3.339%	0.043%	0.001%
5		8.298%	0.001%	0.610%	0.002%	-
6	-	3.078%	-	0.099%	-	-
≥ 7	-	1.498%	-	0.019%	-	-

Table 2: Distribution and examples of classes with different feature types.



POS Tag Sequence	%
NNS IN NNP	10.05%
NN NNS	7.56%
JJ NNS	7.35%
NN NNS IN NNP	4.68%
JJ NNS IN NNP	4.13%
NNP NNS	3.94%
JJ NN NNS	3.48%
CD NNS	2.63%
NN NN NNS	2.61%
NNS IN JJ NNP	2.00%
NNP NN NNS	1.88%
NN NNP NNS	1.88%
NNS IN JJ	1.80%
JJ NNP NNS	1.71%
NN VBD NNS	1.70%
NNS VBD CD	1.64%
NNS	1.58%
NNP NNS NNS	1.50%
NNP NNS IN NNP	1.43%
NNS IN NN	1.40%
NNS IN NNP NN	1.33%
JJ JJ NNS	1.18%
NNS IN NN NN	1.16%
NNS IN NNP NN NNP	1.09%
Long tail	30.29%

Fig. 1: Long tail distribution of POS Tag sequences for Wikipedia categories.

This work concentrates on the use of Wikipedia category links for the analysis and evaluation of NLCDs. The scale, decentralization and domain variety of Wikipedia categories makes it an ideal resource for the investigation of NLCDs under a high variety scenario. We believe that most of the results from this paper can be transported to other complex NLCDs categorization systems. Similar features and patterns can be observed in other examples of complex NLCDs, for example in a domain specific scenario such as the IFRS taxonomy[2] and the US GAAP Taxonomy[3]. Examples of categories are: *Franchised Units*; *Partially Owned Properties*; *Residential Portfolio Segment*; *Assets arising from exploration for and evaluation of mineral resources*; *Key management personnel compensation, other long-term employee benefits.*

[2] http://www.ifrs.org/
[3] http://xbrl.us/taxonomies/

5 Representation Model

5.1 Overview

The representation model is aimed towards facilitating the fine-grained integration between different NLCDs, providing the creation of an integrated and more structured model from the category descriptors. The representation also has an associated interpretation model which aims at making explicit the algorithmic interpretation of the descriptor in the integrated graph.

5.2 Representation Elements

A NLCD can be segmented into 7 representation elements:

- *Entity:* Entities inside a NLCD are terms which are sub-expressions of the original category which can describe predicates or instances. The entities map to a subset of the content words (open class words), which carry the main content or the meaning of a NLCD. Words describing entities can combine *nouns*, *adjectives* and *adverbs*. The entities for an example NLCD *'Snow Or Ice Weather Phenomena'* are *'Snow'*, *'Ice'*, *'Weather Phenomena'*. Entities are depicted as e_i in Figure 2(1).
- *Class & Entity core:* Every entity will contain a semantic nucleus, which corresponds to the phrasal head and which provides its core meaning. For the predicate *'Snow Or Ice Weather Phenomena'*, *'Phenomena'* is the class & entity core. Depicted as '*' in Figure 2(5).
- *Relations:* Relation terms are binary predicates which connect two entities. In the context of predicate descriptors, relation terms map to closed class words and binary predicates, i.e. prepositions, verbs, comparative expressions (*same as*, *is equal*, *like*, *similar to*, *more than*, *less than*). Depicted as p_i in Figure 2(1).
- *Specialization relations:* Specialization relations are defined by the relations between words w_i in the same entity, where w_{i+1} is specialised by w_i. Representing by an unlabelled arrow in Figure 2(4).
- *Operators:* Represents an element which provides an additional qualification over entities as a unary predicate. Operators are specified by an enumerated set of terms which maps to adverbs, numbers, superlative (suffixes and modifiers). Quantifiers: e.g. *one*, *two*, *many (much)*, *some*, *all*, *thousands of*, *one of*, *several*, *only*, *most of*; modal: e.g. *could*, *may*, *shall*, *need to*, *have to*, *must*, *maybe*, *always*, *possibly*; superlatives: e.g. *largest*, *smallest*, *top most*; ordinal: *1st*, *second*. Depicted in Figure 2(2).
- *Conjunctions & Disjunctions:* A disjunction between two elements ($w_i \vee w_{i+1}e_j$) is defined by the distribution of specialization relations: e_j is specialised by w_i and e_j is specialized by w_{i+1}. A conjunction is treated as an entity which names the conjunction of two entities through a conjunction labelled link. The conjunction representation is depicted in Figure 2(2,4).
- *Temporal Nodes:* Consists in the representation of temporal elements references into a normalized temporal range format.

The representation elements previously described are defined below.

Let *Stopwords* be a set of stopwords that are not used in the representation model. For each complex category cl, we associate the set $Terms(cl)$ formed by all **relevant terms of** cl, that is, $Terms(cl) = \{t : t \in (cl \setminus Stopwords)\}$.

The set $Terms(cl)$ can be split into the following disjoint sets:

- *Ent(cl)* is formed by nouns, adjectives and adverbs. The terms in *Ent(cl)* are called **atomic terms** and the elements that provide the core meaning of a complex category are called **term nucleus** and will be denoted by t^*;
- *Rel(cl)* is formed by prepositions, verbs and comparative expressions which represent the relations presented in cl;
- *Oper(cl)* is formed by operators;
- *Temp(cl)* is formed by temporal elements. Temporal elements are normalized into $(dd_i/mm_i/yyyy_i - dd_f/mm_f/yyyy_f)$ representing a time interval starting in $dd_i/mm_i/yyyy_i$ and ending in $dd_f/mm_f/yyyy_f$.

Definition 1. *A complex category cl is represented by a graph $G(cl)$ defined as an injective total function*

$$G(cl) : I \rightarrow 2^{(N \cup I) \times R \times (N \cup I)}$$

where:

- $N = Ent(cl) \cup Oper(cl) \cup Temp(cl)$ *is a set of nodes;*
- I *is a set of identifiers;*
- $R = Rel(cl) \cup Rel_{gen}$*, is a set of relations where* $Rel_{gen} = \{is\_specialized\_by, op, time\}$.

Note that in definition 1, the identifiers in I are used to identify a set of triples instead of individual triples. In a graph $G(cl)$ we can have the following types of triples:

- Basic triple: (x, r, y) such that $x, y \in Ent(cl)$ and $r \in R$
- Reified triple: (x, r, y) such that $x, y \in Ent(cl) \cup I$, with one of them belonging to I, and $r \in R$
- Temporal (basic or reified) triple: $(x, time, y)$ such that $x \in Ent(cl) \cup I$ and $y \in Temp(cl)$
- Operator (basic or reified) triple: (x, op, y) such that $x \in Ent(cl) \cup I$ and $y \in Oper(cl)$

The interpretation of each triple is based on an infinite set U of Universal Resource Identifiers (URIs). Each element $x \in Terms(cl) \cup I \cup Rel_{gen}$ is interpreted as $[[x]] \in U$. Thus a triple $tr = (x, r, y)$ is interpreted as $[[tr]] \in U^3$.

In a graph $G(cl)$, the **complete path** P is the sequence of sets of identifiers $< S_{id_1}, S_{id_2}, \cdots, S_{id_n} >$ such that:

- $\forall id \in S_{id_1}$, id identifies a basic triple $tr = (t^*, r, y)$ where t^* is a term nucleus;

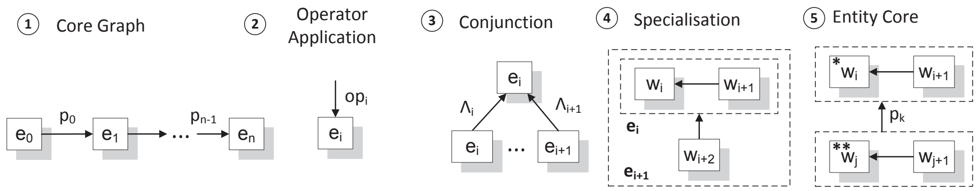


Fig. 2: Graph patterns showing the relations present in the graph representation.

- $\forall id \in S_{id_i}$, all identifiers id' that appear in triples $tr \in id$ are such that $id' \in S_{id_{i-1}}$

Example 1. Consider the complex category cl_1:

20th-centuryRulersOfConstituentOrUnrecognizedStatesInNorthAmerica

The relevant terms $Terms(cl_1)$ are formed by:

- $Ent(cl_1) = \{North, America^*, States^*, Constituent, Unrecognized, Rulers^*\}$
- $Rel(c_1) = \{of, in\}$
- $Temp(cl_1) = \{01/01/1900 - 31/12/2000\}$

Let $I = \{t_1, t_2, t_3, t_4, t_5\}$ be the set of identifiers where:

- t_1 :*NorthAmerica*
- t_2 :*StatesInNorthAmerica*
- t_3 :*ConstituentOrUnrecognizedStatesInNorthAmerica*
- t_4 :*20th-centuryRulers*
- t_5 :*20th-centuryRulersOfConstituentOrUnrecognizedStatesInNorthAmerica*

The graph $G(cl_1)$ is defined as:

- $t_1 = \{(America^*, is\_specialized\_by, North)\}$
- $t_2 = \{(States^*, in, x) \mid x \in t_1\}$
- $t_3 = \{(x, is\_specialized\_by, Constituent), (x, is\_specialized\_by, Unrecognized) \mid x \in t_2\}$
- $t_4 = \{(Rulers^*, time, 1900 - 2000)\}$
- $t_5 = \{(x, of, y) \mid x \in t_4 \text{ and } y \in t_3\}$

and the complete path is $P =< \{t_1, t_4\}, \{t_2\}, \{t_3\}, \{t_5\} >$

Further examples are depicted in Figure 3. The representation can be directly translated into an Resource Description Framework (RDF) graph. Most of the overhead in the translation is due to the fact that words mapping to classes need to be instantiated and later reified. Terms which are classes and which need to be reified are reflected as instances. Figure 4 shows an example of the corresponding RDF representation for the *'Tennis Players At 1996 Summer Olympics'* category. The example shows the alignment of the corresponding WordNet senses and DBpedia instances.

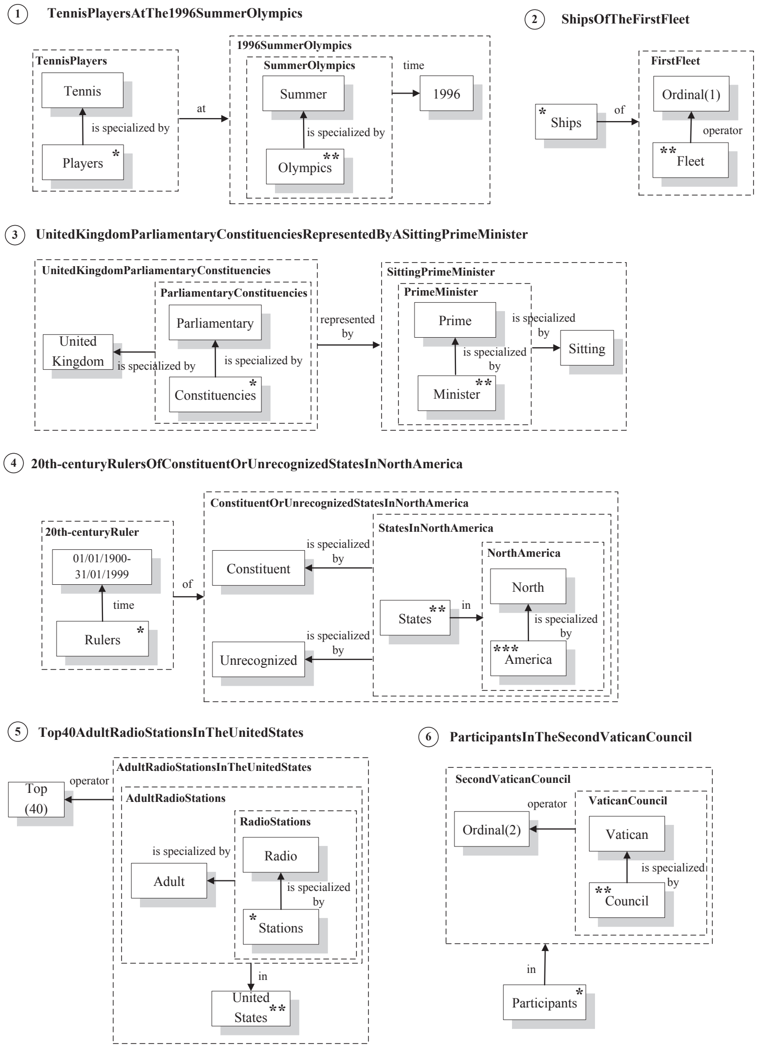


Fig. 3: Categories following the representation model.

6 Extraction

This section describes the process for extracting NLCDs into the proposed representation model. Figure 5 shows the components and the extraction workflow. The NLCD extraction consists of the following steps:

1. **POS Tagging**: Detection of the lexical categories of the NLCD words. The extractor uses the NLTK POS Tagger[4].

[4] http://www.nltk.org

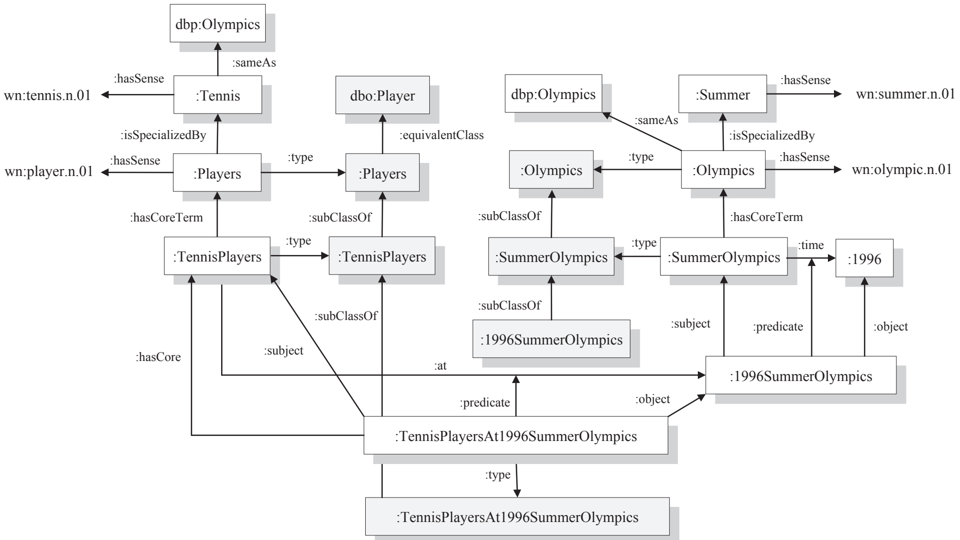


Fig. 4: RDF representation of the category *'Tennis Players At 1996 Summer Olympics'*. Classes are in light gray, while instances are represented in white.

2. **Segmentation**: The segmentation of the NLCD starts by detecting the relations and splitting the descriptor into a set of entities and relations.
3. **Entity Detection**: This step consists on the detection of 3 types of entities: named entities, operators and temporal references:
   (a) *Named entities*: The detection of named entities is based on the creation of a gazetteer from DBpedia 3.9 instances. Elements tagged as nouns and proper nouns are checked against the gazetteer.
   (b) *Operators*: Operators are detected using the combination of an enumerated list of operators and regular expressions based on POS Tags.
   (c) *Temporal References*: Temporal references are detected using regular expressions. At this step temporal references are normalized.
4. **Specialization ordering**: This step consists in defining the specialization sequence for the terms inside each entity. Two heuristic indicators are used in the determination of the ordering of the terms inside the classes: POS Tags and a corpus-based specificity measure (inverse document frequency (IDF) over Wikipedia 2013 text collection). The POS Tags are used to segment the terms in the class descriptor into a coarse-grained word ordering based on the lexical categories. The ordering is defined by the relations (NN - is specialised by → JJ, JJ - is specialised by → RB). For an entity containing words from the same lexical category, IDF is used to define the ordering: Lower IDF - is specialized by → Higher IDF.
5. **Word Sense Disambiguation**: Let the sequence of words $w_0, w_1, ..., w_n$ be the natural language descriptor for a category c. Let $g_0, g_1, ..., g_k$ be the WordNet glosses associated with the senses for w_i for $0 \leq i \leq n$. Let $\kappa(w_i)$

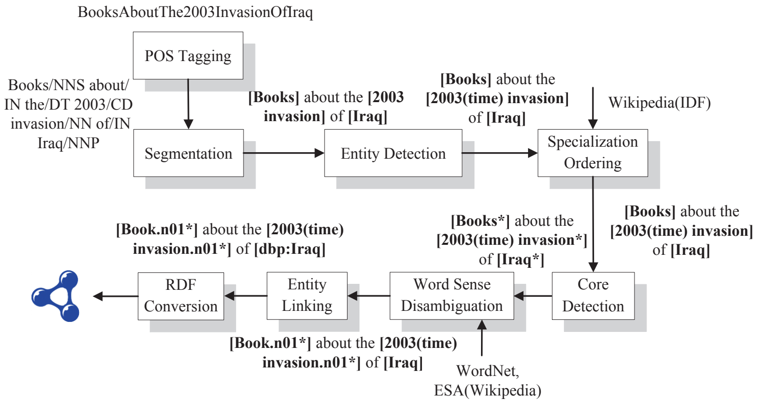

BooksAboutThe2003InvasionOfIraq
POS Tagging
Books/NNS about/ IN the/DT 2003/CD invasion/NN of/IN Iraq/NNP
Segmentation
[Books] about the [2003 invasion] of [Iraq]
Entity Detection
[Books] about the [2003(time) invasion] of [Iraq]
Wikipedia(IDF)
Specialization Ordering
[Books] about the [2003(time) invasion] of [Iraq]
Core Detection
[Books*] about the [2003(time) invasion*] of [Iraq*]
Word Sense Disambiguation
WordNet, ESA(Wikipedia)
[Book.n01*] about the [2003(time) invasion.n01*] of [Iraq]
Entity Linking
[Book.n01*] about the [2003(time) invasion.n01*] of [dbp:Iraq]
RDF Conversion

Fig. 5: Extraction components and workflow.

be the context of w_i defined by $w_0, w_1, ..., w_n \setminus w_i$. The sense for w_i is given by $sr_{ESA}(\kappa(w_i), g_j)$ where sr_{ESA} is the distributional semantic relatedness measure (Explicit Semantic Analysis) between the WordNet glosses and the category context.

6. **Entity linking**: Entity linking uses DBpedia as a named entity base and a ranking function based on TF/IDF over labels, entity cardinality and levehnstein distance.
7. **RDF conversion**: At this point the relations are represented as an internal set of extracted graphs following the proposed representation model. The model is then converted into an RDF graph.

7 Evaluation

The extraction approach was evaluated by randomly selecting a sample of 2,696 Wikipedia categories from the original set of 287,957 categories. These categories were extracted and manually evaluated according to eight extraction features: *entity segmentation*, *relation extraction*, *unary operators*, *specialization sequence*, *detection of class core*, *detection of entity core*, *word sense disambiguation* and *entity linking*. The features map to the components of the extraction approach. Table 3 shows the accuracy for each feature.

The low error in entity segmentation, relation extraction and specialization sequence shows the generality of the extraction rules in relation to the tractable subset of natural language category descriptors. Additionally, the high accuracy in the determination of the sequence of specialization relations, detection of class and entity cores shows the correctedness in the construction of the taxonomic structure. For an open domain scenario, the WSD approach based on Explicit

	Entity Segmentation	**Relations**	**Unary Operators**	**Specialization Relations**	**Class Core**	**Entity Core**	**WSD**	**Entity Linking**
Accuracy	79.38%	95.96%	99.74%	97.81%	99,37%	81.86%	82.2%	78.1%

Table 3: Accuracy for each extraction feature.

Semantic Analysis achieved an average accuracy of 82,2%. Domain specific categories may require more constrained distributional semantic models.

Additionally, the graph extraction time was evaluated with regard to the extraction performance time. The experiment was carried in a i5-3317U (1.70GHz) CPU computer with 4GB RAM (4 core, 2 threads per core). The extraction was evaluated with regard to three main categories: (i) graph extraction time (**9.8 ms per graph**), (ii) word sense disambiguation **121.0 ms per word** and (iii) entity linking **530.0 ms per link**. Entity Linking is the most expensive operation, followed by Word Sense Disambiguation. The overall extraction time per NLCD shows that the approach can be integrated into medium-large scale categorization tasks. Each category generates an average of 10.2 RDF triples. The extraction tool is available at the website[5].

8 Conclusion & Future Work

This paper analyses the use of complex natural language category descriptors (NLCDs) and proposes a representation model and an extraction approach for NLCDs. The accuracy of the proposed approach was evaluated over Wikipedia category links, achieving an overall structural accuracy above 78%. Future work will focus on the evaluation of the approach under domain-specific NLCDs.

References

1. Suchanek, F. Kasneci, G., Weikum, G.: YAGO: A Large Ontology from Wikipedia and WordNet, In Proc. of the 16th International Conference on World Wide Web, pp. 697-706 (2007).
2. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In Proc. of the Intl. Joint Conference On Artificial Intelligence (2007).
3. Specia, L., Motta, E.: Integrating Folksonomies with the Semantic Web: In Proc. of the 4th European Conference on the Semantic Web, pp. 624-639, (2007).
4. Cattuto C., Benz, D., Hotho, A., Stumme, G.: Semantic grounding of tag relatedness in social bookmarking systems. In Proc. 7th Intl. Semantic Web Conference (2008).
5. Limpens, F., Gandon, F., Buffa, M.: Linking Folksonomies and Ontologies for Supporting Knowledge Sharing: a State of the Art, Technical Report (2009).
6. Voss, J.: Linking Folksonomies to Knowledge Organization Systems, Communications in Computer and Information Science v. 343, 2012, pp 89-97.

[5] http://graphia.dcc.ufrj.br/nlcd

7. Cimiano, P., Handschuh, S., Staab, S.: Towards the Self-Annotating Web, In Proc. of the 13th International Conference on World Wide Web, pp. 462-471 (2004).