Towards a General Entity Representation Model

Barbara Bazzanella, Junaid Ahsenali Chaudhry, Themis Palpanas, Heiko Stoermer

University of Trento
b.bazzanella@email.unitn.it, chaudhry@disi.unitn.it,
themis@disi.unitn.eu, stoermer@disi.unitn.it

Abstract. In recent years, there is an increasing interest in the Semantic Web and the relevant technologies, which can have a significant impact in the context of information and knowledge management. An important observation is that the entity identification problem lies at the core of many semantic web applications and the intrinsic difficulties of this problem have hindered progress in this area.
In this paper, we argue for an infrastructure responsible for assigning and managing unique identifiers for entities in the semantic web, and we propose a conceptual model for the storage and management of these entities. The proposed model is generic and flexible and it allows for efficient and effective retrieval and analysis of the stored entities. We discuss the requirements with respect to creating and modifying these entities, as well as to managing their evolution over time. Finally, we study some enhancements of the entity representation, and we discuss the beneficial impact they can have on the performance of the system.

1 Introduction

One of the major problems that have emerged in the Semantic Web (SW) effort is the problem of uniquely identifying entities[1] [1]. Entities play a major role for the SW since they represent the atomic objects of reference and reasoning. Nevertheless, we currently face the problem of identifying and referencing these entities, since different users, or systems, assign different identifiers to the same real-world entities. As a result, we cannot effectively reason about these entities, exactly because they are not consistently being assigned the same identifier.

The entity identification problem is also relevant to information and knowledge management in an enterprise environment. Its successful solution can help enterprises consolidate and integrate all the data about a single entity that are scattered across data sources both inside and outside the boundaries of the enterprise, thus, delivering significantly richer knowledge management opportunities.

It has been argued that the entity identification problem is at the core of the semantic web effort [2]. Along with the problem of assigning global identifiers to entities in the semantic web also come the problems of managing these identifiers

[1] In the rest of this paper, we will use the term entity to refer to individuals, particulars, and instances, as opposed to classes or concepts.

throughout the entire lifetime of the entities. Giving efficient solutions to the above issues is the goal of the *OKKAM Entity Name System (ENS)* [1], a web-scale system for assigning and managing unique, global identifiers to entities in the WWW.

In this study, we focus on the problem of how to efficiently and effectively represent an entity in the context of such a system. We examine the requirements of entity representation on the flexibility of the representation and the functionality of the entire system, and we propose a conceptual model to this effect. We also perform a systematic study about the description of entities, and propose guidelines that can improve the representation and matching of entities through the use of default (i.e., suggested) attributes. To the best of our knowledge, this is the first comprehensive study in these directions.

1.1 Related Work

Douglis et al. [3] propose a storage infrastructure that effectively takes into account not only disk read and writes, but also data creation and deletion. Various techniques that employ different strategies have been proposed for efficiently storing different versions of data objects [4, 5]. Versioning has also been studied in the context of semi-structured documents [6], and efficient query answering algorithms have been proposed [7]. When entities are created and modified, we are interested in keeping track of information related to the provenance of the entity data stored in the repository [8]. Efficient techniques for storing and querying such metadata have been studied in the literature [9, 10]. Chapman et al. [11] propose efficient strategies for reducing the provenance storage size. Several works have focused on the important problems of record linkage and record matching [12–14]. Other studies have focused on the problem of how to efficiently support the above operations in the context of relational database systems [15, 16]. Duplicate detection through record linkage has also been studied [17]. These approaches are based on different flavors of clustering algorithms. Benjelloun et al. [18] propose three algorithms for solving the entity resolution problem, namely, G-Swoosh, R-Swoosh, and F-Swoosh. These algorithms take into account the characteristics of the match and merge functions, and can also provide approximate results.

1.2 Background

We now give a brief overview of the *ENS* (a more detailed presentation can be found elsewhere [1]), which we will use as the basis for our discussion. Note however, that our discussion is relevant to any system for entity identification management. The overall goal of the *ENS* is to handle the process of assigning and managing unique identifiers for entities in the WWW. These identifiers are global, with the purpose of consistently identifying a specific entity across system boundaries, regardless of the place in which references to this entity may appear (see Figure 1).

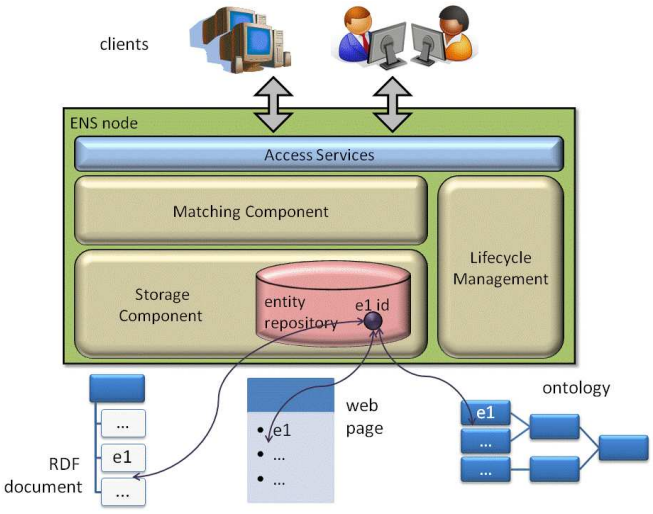

Fig. 1. Schematic of the *ENS* and its interactions.

The *ENS* has a repository for storing entity identifiers (note that this repository will be distributed and replicated) along with some small amount of descriptive information for each entity. The purpose of storing this information is to use it for discriminating among entities, not exhaustively describing them. Entities are described by a number of attribute-value pairs, where the attribute names and the potential values are user-defined (arbitrary) strings, as we will discuss in more detail in Section 2.1.

Clients interact with the system through the *Access Services* layer. Clients can be both human users and applications, and may inquire about the identifier of an entity by providing a set of attributes that describes this entity. If the entity exists in the repository, the system returns its identifier. Clients may also modify the state of the repository, either by inserting a new entity in the system, in which case the *ENS* returns the newly assigned identifier, or by changing some of the attributes of an existing entity. As shown in Figure 1, the end result is that all instances of the same entity (i.e., mentioned in different systems, ontologies, web pages, etc.) are assigned the same identifier. Therefore, joining these documents and merging their information becomes a much more simple and effective process than before.

2 Entity Representation

In this section we discuss the requirements and the design of the entity representation model for the *ENS*.

The most important requirement is that we design an entity representation conceptual model that takes into account the need for flexibility and generality

in the description of entities. Note that there is no fixed schema for the representation of entities, which is essential for ensuring that the *ENS* can represent arbitrary entities, and also for enabling easy access methods to clients that are oblivious about the specific entity representation choices.

Evidently, entities may change as a function of time. The description of these entities in the *ENS* repository should be able to follow this natural evolution, by modifying, adding, and deleting attributes. Similarly to the entities themselves, the entire repository may evolve over time. In this case, we need to address the issues of entity merges and splits. Two entities may merge if we discover that they refer to the same real world entity. A split may occur if we discover that a single entity refers to two real world entities.

When entities are created and modified, we are interested in keeping track of the information related to the provenance of the entity data stored in the repository. The above information can potentially be useful for other algorithms operating on the entities in the repository, such as matching and merging.

Various techniques and algorithms are needed in order to analyze and mine the wealth of information that can be gathered by observing the behavior of the *ENS*, and assist in automating the evolution functions discussed above. This information may come from observing the access and usage patterns of clients as they interact with the *ENS* and explore the entities stored in the repository. Given the amount of information and the real time requirements in the context of the *ENS*, these analysis algorithms should be able to operate in an online fashion, be flexible enough to allow effective and efficient data analysis of the incoming data streams [19, 20], and evolve over time by supporting time-decaying representations of the streaming data [21].

2.1 Entity Representation Conceptual Model

Based on the above requirements, we now present in detail the conceptual model for entity representation in the *ENS*.

In the *ENS*, we represent an entity E as a tuple $E =< oid,\ EQoid,\ Aid,\ prid,\ D,\ M^E >$, with the following information.

- An entity identifier, *oid*, assigned by the system.
- A set of equivalent *ENS* entity identifiers, *EQoid*. This field is used in the case where we discover that two (or more) different entities in the *ENS* refer to the same real world entity. By listing the other *ENS* identifiers in this field, we establish an identity connection among the corresponding entities.
- A set of alternative ids, *Aid*. These are identifiers that other systems (external to the *ENS*) have assigned to the same entity. By listing these identifiers in this field, we establish an identity connection among these identifiers.
- A preferred id, *prid*. This is the identifier that the entity prefers to be known by, and this is the identifier that the *ENS* will return in response to an id query about the entity. The default value for this field is the *ENS* identifier, *oid*.
- A description of the entity, D, which we discuss in more detail below.
- Metadata for the entity, M^E=<M^E_G,M^E_S,$M^E_P M^E_A$>. M^E_G represents the general metadata for the entity, including creation time. M^E_S is the statistical metadata,

including last modification time, number of times the entity was matched, number of times it was selected, and last time it was selected. M_P^E is the provenance metadata for the entity, that is, the source of information (where we found this piece of information), and the agent that produced these data (name of software, manual process, etc.). Finally, M_A^E is the access control metadata for the entity.

We now move to D=<t,A,R>, which contains all the information that describes the entity:
- The semantic type of the entity, t.
- The set of attributes describing the characteristics of the entity, A.
- The set of external references that refer to this entity, R.

We should note that the semantic type field, t, can be very useful for entity matching, since it provides some additional information for classifying and identifying the entity. At the same time though, our goal is to keep the entity representation model as simple and general as possible, which means that this field should not take values from the domain of a fully-fledged semantic ontology (note that in that case, we would have to maintain the ontology itself, too!). We describe a viable and effective approach for this problem in Section 3. We also take into account the fact that attribute names are arbitrary (since they are user-defined), and extend our approach to address this issue as well. In the next paragraphs, we discuss in more detail the attribute and reference fields of D.

An attribute, A, is a tuple of the form A = <n, v, *veid*, MA>, containing the following information: the name of the attribute, n, the value of the attribute, v, the entity identifier assigned by the *ENS* for the entity described by v, *veid*, and the metadata for attribute A, M^A. M^A=<M_G^A,M_S^A,$M_P^A M_A^A$> refers to the metadata of the specific attribute A of a specific entity E. The tuple M_G^A is the general metadata for the attribute, including creation time, and natural language of the name/value pair. M_S^A is the statistical metadata for the attribute, including last modification time, number of times the attribute was used in a query, and the last time it was used in a query. M_P^A is the provenance metadata for the attribute that includes the source of information, and the agent that produced these data. M_A^A is the access control metadata for the attribute.

A reference R is a tuple of the form R = <c, p, M^R >, containing the following information: the category of reference (e.g., ontology), the URL pointing to the external reference, and the metadata for the reference, M^R.
M^R=<M_G^R,M_S^R,$M_P^R M_A^R$> refers to the metadata of a specific reference R of a specific entity E. These metadata are as follows. M_G^R is the general metadata for the reference that includes the creation time. M_S^R is the statistical metadata for the reference, including the last time the reference was checked (i.e., latest time we know this reference was valid). M_P^R is the provenance metadata for the reference that includes the source of information, and the agent that produced these data. Finally, M_A^R is the access control metadata for the reference.

Before we conclude the discussion of the entity representation model, we would like to emphasize the significance of the metadata. These metadata can play an important role in query answering and ranking, as well as in the management and evolution of the entity repository over time. For example, assigning

weights to attributes according to the provenance and statistical metadata can help improve the performance of entity matching. Moreover, the information on usage patterns that can be derived from the statistical metadata can drive automatic algorithms for the self-management and evolution of the repository.

3 Defining Default Attributes: a Bottom-Up Approach

Although in the *ENS* there is no fixed schema of entity types and attributes to be used for describing entities, we aim to encourage some homogeneity in the description of entity profiles. For this purpose we attempt to encourage the clustering of entities into a small set of types and provide suggestions for the corresponding descriptions by means of a default set of attributes for each entity type. By following this approach we hope to improve the functionality of the system in at least three different levels. First, having information about the types of entity can be useful to guide the selection of specialized entity matching algorithms. Second, the matching techniques can be considerably simplified and improved. Last, the efficiency of the matching process can be improved, by clustering attributes that convey overlapping information and reducing the number of attributes to consider. In order to derive such a default schema we decided to adopt a bottom-up approach, performing an experiment as follows.

Selection of Top Level Categories: The first step in the experiment was to select an appropriate collection of top-level categories. In order to obtain a reasonably exhaustive, but at the same time limited, set of categories we adopted a top-down approach. We analyzed the main top-level ontologies available in literature (Wordnet [23], Dolce [22, 24], Sumo [25], Cyc [26]) to integrate important ontological distinctions from these ontologies. The goal was to identify a set of few categories to use as a test-bed on which to perform the experimental investigation. A detailed analysis of the procedure is reported in [27]. At the end of our analysis we identified the following six top-level categories: *Person*, *Organization*, *Event*, *Artifact*, *Location*, and *Other*[2].

Methodology: After establishing the top level categories for our study, we conducted a user-study. Our aim was to investigate which attributes are more frequently reported by people when describing the selected entity types, and to derive lists of default attributes for those entity types.

In order to get subjects to generate a set of representative attributes, we adopted the feature-listing task paradigm [28]. Following this paradigm, we asked subjects to produce lists of attributes they think relevant to identify uniquely members of our categories. Since our top level categories were at a high level of abstraction, we decided to introduce a certain number of subcategories for each of them in addition to the simple top-level category (named "neutral category"). By means of this expedient we could identify a core set of attributes shared by the different subcategories within the same top-level category. In Table 1 we report the lists of subcategories used in the experiment.

[2] We point out that the last category, *Other*, is a miscellaneous category that contains all entities that are not classifiable in one of the other categories.

Person	Organization	Event	Artifact	Location
politician	company	conference	product	tourist location
manager	association	meeting	artwork	city
professor	university	exhibition	building	shop
sports person	government	show	book	hotel
actor	agency	accident	article of clothing	restaurant
person	organization	event	object	location
		sports event		

Table 1. Top-level Categories and Subcategories.

Implementation and Subjects: The experiment was conducted with a between-subjects design. That is, each subject was randomly assigned to only one combination of 5 scenarios (one subcategory for each top level category). This was required so as to eliminate interference between different scenarios. The experiment was conducted through the web in three different versions: English (eng), Italian (it), and Chinese (chi). We collected data from 358 participants (159 for the English version, 194 for the Italian version and 5 for the Chinese version[3]).

Normalization: The linguistic nature of the task was bound to account for a certain degree of variability in our data. To deal with this variability we normalized the data with a semi-automatic procedure, converting the entries of our database in a standard form. After having removed all typing errors, we reported the attributes in a unique morphological form, removing articles, normalizing the use of prepositions and the singular-plural inflections, as well as fixing the order for composed attributes. Finally we aggregated attributes characterized by semantic overlaps (such as synonyms).

Measures and Results: The problem of suggesting descriptions for types of entities at a high level of abstraction corresponds to identifying a set of general attributes used by subjects across the subcategories of the same top-level category. When aggregating the data from these subcategories, we require a measure to evaluate the importance of an attribute f for the top-level category c.

The first measure that we examined was the dominance measure, that can be formalized as a function ϕ: $C \times F \rightarrow N$:

$$dominance = \phi(c, f) = |\{s \in S : f \in F_s^c\}|$$

where S is the sample of subjects, and F_s^c is the set of attributes listed by the subject s given the category c. In other words, the dominance ϕ of the attribute f for the category c corresponds to the number of subjects that reported the attribute f for the category c.

Note that the dominance measure does not guarantee that in the first positions of the ranked list (in descending order of dominant attributes) appear attributes shared between the subcategories, because we have aggregated the data from the subcategories. If an attribute is reported by all participants for

[3] Because of the limited number of participants in the Chinese version, we present the results only of the Italian and English versions.

a specific subcategory (e.g., "political party" for *politician*) and only for this subcategory, it is possible that this attribute appears among the first attributes for the corresponding top level category (e.g., *Person*). In order to derive a set of default attributes that are both frequently reported by subjects for a specific top-level category, and also highly shared across the subcategories within the same top-level category, we used a second measure, *local sharedness*, that quantifies the level of sharing of an attribute f across a collection of subcategories:

$$sharedness_{loc} = \psi_l(f) = \frac{|S[f]|}{|S_c|}$$

where $|S[f]|$ is the collection of the subcategories that have in common the attribute f, and $|S_c|$ is the collection of all subcategories belonging to the category c. Weighting the measure of dominance by local sharedness, $\phi_w = \psi_l * \phi$, we obtained the list of default attributes for our top-level categories. The complete list of these attributes is reported in Table 2.

3.1 Applications of Default Attributes Analysis

We now sketch two possible uses of the default attributes reported in Table 2, which demonstrate the importance of the above analysis.

One possible application is the suggestion of a default schema for adding new entities in the system. Client applications can be designed to give the user a selection of our top-level categories, in order to get the class of the new entity to be added. After the user has selected the entity type, the system can propose the list of default attributes that our analysis found to be the most representative for the specific entity type. At this point, the user can fill-in the values for these attributes (or for some of them, or even insert new attributes) to describe the entity to be created. The result is a relatively uniform representation for these entity types, which can prove very beneficial for the entity matching process.

A second application is entity disambiguation. When a new entity is added to the repository, it is necessary to verify if it is already stored in the system. If the entity type of the new entity is not known, our analysis can help improve the performance of the entity disambiguation algorithms. For example, our analysis showed that the attribute "surname" is one of the most important attributes to describe a *Person*. If we identify that such an attribute is present in the new entity description, then entity matching can be limited to (or start from) entities of the entity type *Person*.

4 Conclusions and Future Work

In this paper, we argue for an entity naming system, where unique identifiers for entities are assigned and managed. We examine the special requirements of representing entities in such a system, propose a model for entity representation, as well as a technique for enhancing its quality, and describe how the proposed approach can help achieve the overall goals.

	English		Italian	
Category	**Attributes**	ϕ_w	**Attributes**	ϕ_w
Person	name	110	nome (name)	89
	age	49	etá (age)	73
	gender	44	cognome (surname)	64
	birth-date	29	tipo (type)	56
	surname	24	data di nascita (birth-date)	34
	education	24	esperienze (experiences)	29
	country	20	titolo di studio (education)	22.5
		N= 145		N=171
Organization	name	77	nome (name)	87
	location	37	tipo (type)	54
	country	34	scopo/i (aim/s)	44
	address	31	luogo (location)	37.5
	type	23	sede (head office)	15.83
	size	12	settore (sector)	15.33
	Web site url	11.66	indirizzo (address)	12.5
		N= 137		N=168
Event	location	116	luogo (location)	126
	date	69	data (date)	74
	time	64	tipo (type)	68
	name	49	ora (time)	57
	participants	40	partecipanti (participants)	33.42
	type	26	durata (duration)	28.28
	duration	18	argomento (topic)	21.71
		N= 146		N=161
Artifact	color/s	46	colore/i (color/s)	74
	name	33	tipo (type)	60
	size	29.16	dimensione/i (dimension/s)	36
	type	28	materiale (material)	35
	price	20.83	prezzo (price)	28.33
	material	20	autore (author)	25
	shape	16	luogo (location)	20
		N= 140		N=168
Location	name	86	luogo (location)	78
	country	50	nome (name)	73
	location	48	tipo (type)	57
	address	39.1	coord. geo. (geo coordinates)	29.1
	geo coordinates	35.83	indirizzo (address)	18.66
	city	25.83	stato (country)	15.83
	price	14.86	numero di abitanti (number of citizens)	14.5
		N= 145		N=169

Table 2. Attributes ranked according to weighted-dominance.

We are currently working in two research directions. First, to improve the suggested default attributes. In our analysis, we decided to reduce as much as possible the semantic aggregation of attributes. However, we still need to limit the overlap of attributes. To this effect, we plan to perform a post-processing analysis. One idea is to follow a simple criterion for aggregating attributes: when two (or more) attributes share partial content, the attribute that conveys more information will incorporate the other(s). For example, the attribute "age" is subsumed by the attribute "birth-date", and can therefore be incorporated into the "birth-date".

Second, to investigate the impact of the proposed approach on the quality of the system, for example, in terms of precision in entity matching. We are currently in the process of conducting large-scale experiments in order to test the behavior and performance of our solution.

Acknowledgments This work was partially supported by the FP7 EU Large-scale Integrating Project OKKAM - Enabling a Web of Entities (contract no. ICT-215032). For more details, visit http://www.okkam.org.

References

[1] Bouquet, P., Stoermer, H., Bazzanella, B.: An entity name system (ens) for the semantic web. In: ESWC. (2008) 258–272
[2] Bouquet, P., Stoermer, H., Niederee, C., Mana, A.: Entity Name System: The Backbone of an Open and Scalable Web of Data. In: ICSC 2008. Number CSS-ICSC 2008-4-28-25 (2008)
[3] Douglis, F., Palmer, J., Richards, E.S., Tao, D., Tetzlaff, W.H., Tracey, J.M., Yin, J.: Position: short object lifetimes require a delete-optimized storage system. In: ACM SIGOPS European Workshop. (2004) 6
[4] Santry, D.S., Feeley, M.J., Hutchinson, N.C., Veitch, A.C., Carton, R.W., Ofir, J.: Deciding when to forget in the elephant file system. In: SOSP. (1999)
[5] Mahalingam, M., Tang, C., Xu, Z.: Towards a semantic, deep archival file system. In: FTDCS. (2003)
[6] Chien, S.Y., Tsotras, V.J., Zaniolo, C.: Efficient schemes for managing multiversionxml documents. VLDB J. **11** (2002)
[7] Chien, S.Y., Tsotras, V.J., Zaniolo, C., Zhang, D.: Supporting complex queries on multiversion xml documents. ACM Trans. Internet Techn. **6** (2006)
[8] Tan, W.C.: Provenance in databases: Past, current, and future. IEEE Data Eng. Bull. **30** (2007)
[9] Velegrakis, Y., Miller, R.J., Mylopoulos, J.: Representing and querying data transformations. In: ICDE. (2005)
[10] Srivastava, D., Velegrakis, Y.: Intensional associations between data and metadata. In: SIGMOD. (2007)
[11] Chapman, A., Jagadish, H.V., Ramanan, P.: Efficient provenance storage. In: SIGMOD. (2008)
[12] Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S.: Duplicate record detection: A survey. TKDE **19** (2007)
[13] Sarawagi, S., Bhamidipaty, A.: Interactive deduplication using active learning. In: KDD. (2002)
[14] Galhardas, H., Florescu, D., Shasha, D., Simon, E., Saita, C.A.: Declarative data cleaning: Language, model, and algorithms. In: VLDB. (2001)
[15] Koudas, N., Marathe, A., Srivastava, D.: Flexible string matching against large databases in practice. In: VLDB. (2004)
[16] Guha, S., Koudas, N., Marathe, A., Srivastava, D.: Merging the results of approximate match operations. In: VLDB. (2004)
[17] Dong, X., Halevy, A.Y., Madhavan, J.: Reference reconciliation in complex information spaces. In: SIGMOD. (2005)
[18] Benjelloun, O., Garcia-Molina, H., Menestrina, D., Su, Q., Whang, S.E., Widom, J.: Swoosh: a generic approach to entity resolution. VLDB J. **accepted for publication** (2009)
[19] Palpanas, T., Kalogeraki, V., Gunopulos, D.: Online distribution estimation for streaming data: Framework and applications. In: SEBD. (2007) 430–438
[20] Tantono, F.I., Manerikar, N., Palpanas, T.: Efficiently discovering recent frequent items in data streams. In: SSDBM. (2008)
[21] Palpanas, T., Vlachos, M., Keogh, E.J., Gunopulos, D.: Streaming time series summarization using user-defined amnesic functions. TKDE **20** (2008)
[22] Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L.: Sweetening ontologies with dolce. In: EKOW. Volume 2473. (2002)
[23] Oltramari, A., Gangemi, A., Guarino, N., Masolo, C.: Restructuring wordnet's top-level: The ontoclean approach. In: OntoLex. (2002)
[24] Masolo, C., andA. Gangemi, S.B., Guarino, N., Oltramari, A.: WonderWeb Deliverable D18 Ontology Library (final). (2003)
[25] Niles, I., Pease, A.: Towards a standard upper ontology. In: FOIS. (2001)
[26] Matuszek, C., Cabral, J., Witbrock, M., DeOliveira, J.: An introduction to the syntax and content of cyc. In: AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering. (2006)
[27] Bazzanella, B., Stoermer, H., Bouquet, P.: Top Level Categories and Attributes for Entity Representation. Technical Report 1, University of Trento (2008) `http://eprints.biblio.unitn.it/archive/00001467/`.
[28] McRae, K., Cree, G., Seidenberg, M., McNorgan, C.: Semantic feature production norms for a large set of living and nonliving things. Behavior Research Methods, Instrument and Computers **37** (2005)