# Mapping Natural Language Labels to Structured Web Resources

Valerio Basile[1], Elena Cabrio[2], Fabien Gandon[2], and Debora Nozza[3]

[1] University of Turin, Italy
basile@di.unito.it
[2] Université Côte d'Azur, Inria, CNRS, I3S, France
elena.cabrio@unice.fr,fabien.gandon@inria.fr
[3] University of Milano-Bicocca, Milan, Italy
debora.nozza@disco.unimib.it

**Abstract.** Mapping natural language terms to a Web knowledge base enriches information systems without additional context, with new relations and properties from the Linked Open Data. In this paper we formally define such task, which is related to word sense disambiguation, named entity recognition and ontology matching. We provide a manually annotated dataset of labels linked to DBpedia as a gold standard for evaluation, and we use it to experiment with a number of methods, including a novel algorithm that leverages the specific characteristics of the mapping task. The empirical evidence confirms that general term mapping is a hard task, that cannot be easily solved by applying existing methods designed for related problems. However, incorporating NLP ideas such as representing the context and a proper treatment of multiword expressions can significantly boost the performance, in particular the coverage of the mapping. Our findings open up the challenge to find new ways of approaching term mapping to Web resources and bridging the gap between natural language and the Semantic Web.

## 1 Introduction

Words, labels[4], multiword expressions and keywords are used, among other things, to summarize the topic of articles, to index documents, to improve search, to organize collections and to annotate content in social media. Because of their ubiquity, disambiguating and improving the processing of natural language terms has an immediate and important impact on the performances of many information systems.

Being able to map sets of terms to linked data resources contributes to create new interoperable resources and to transfer knowledge across different applications. Take for example a large, carefully crafted ontology such as KnowRob [14], a framework for robotic perception and reasoning. Despite its wide usage among robotic applications, the basic concepts of KnowRob (objects, places, actions)

[4] Throughout the paper, the words *label* and *term* are used somewhat interchangeably, although we typically refer to labels as terms attached to other entities, e.g. images.

are labeled by arbitrary strings (keywords), isolated from general Web of Data knowledge bases. Such mapping would enrich the original resource with new relations and properties, and the Linked Data cloud with new, often carefully crafted information. Indeed, recent work in robotics highlights the need for linked data resources as a source of background knowledge for domestic robots [6, 16].

In this work, we explore the task of linking sets of labels from an arbitrary resource to a structured knowledge base, a problem we consider at the crossroad between *word sense disambiguation* and *ontology matching*. The contributions of this paper are: i) a formal definition of the term mapping task, ii) a use case scenario, where the labels of a resource from computer vision are linked to a general-purpose Web resource, iii) a large, manually annotated dataset of objects and locations linked to DBpedia, iv) a novel method for solving the mapping task and a benchmark for its evaluation.

The problem is related to *entity linking*, that is, the task of detecting entities in a segment of natural language text and linking them to an ontology. The main differences are that 1) the terms to link are already given, and 2) there is no context for the terms to disambiguate, which are instead given simply as a list. With respect to the second point, we can alternatively state that the set of keywords is itself the context, in the sense that it could give, as a whole, helpful hints for the disambiguation of the single keywords.

Formally, the problem is defined as follows:

*Given an input set of terms $K = k_1, ..., k_n$ and a target knowledge base $S \subseteq (R \setminus L) \times P \times R$, where $R$ is a set of resources, $P \subseteq R$ is the subset of properties, and $L \subseteq R$ is the subset of literals. The goal of the task is that of defining a mapping function $f : K \to R$.*

Two constraints can be optionally posed on the target mapping function: a *total* function, that is, defined on the entire input set, would yield a mapping where every input term is associated to a resource in the target knowledge base. This could be useful in scenarios where the robustness of the mapping is more important than its accuracy. One could also want the mapping function to be *injective*, that is, no pair of distinct input terms is mapped to the same resource, for example in applications where it is known *a priori* that the terms refer to distinct entities.

Depending on the application scenario, it also makes sense to constrain the set of candidate resources in the target knowledge base. For example, we may want to link a set of terms only to instances, rather than classes, or just properties, leaving out other type of resources.

The rest of the paper is structured as follows. We first give an overview of problems and methods related to the term mapping task (Section 2), then we introduce a number of methods to solve it (Section 3). We introduce a relevant use case and test the approaches on a newly created benchmark (Section 4). Finally, we discuss the results (Section 5) and lay down plans to approach term mapping to Web resources in future work (Section 6).

## 2 Related Work

Linking terms to Web resources may find its application in several tasks. Augenstein et al. [1] explicitly approach the task of "mapping keywords to Linked Data resources", with the main goal of producing better queries on linked data resources. In their work, the authors propose a supervised method to map the keywords in natural language queries to *classes* in linked data ontologies. This work served as inspiration to us to formalize the term mapping problem, while differing from its general case, where the keywords are mapped to linked data *resources* and not only classes. Freitas et al. [5] present an overview of approaches to querying linked data, where the subtask of *entity reconciliation* is basically a term mapping step, and two main families of approaches are studied, respectively coming from information retrieval and natural language processing. The former solutions exploit linked data relations such as `owl:sameAs` to facilitate the mapping of disambiguated keywords, while the latter approaches leverage lexical resources and their network structure to link words to semantic entities.

When dealing with flat lists of terms, a relevant task is that of inferring some kind of structure among them. This problem can be approached by linking terms to classes in an ontology first, in order to exploit the relations between classes. Limpens [7], for instance, reports the need of solving term mapping-related issues such as accounting for spelling variations and measuring the semantic similarity of tags from a folksonomy in order to link them to an ontology on the Web. Methods for solving such intermediate tasks, e.g, as described in Specia and Motta [13] and Damme et al. [4], could be directly integrated into a method for term mapping. Meng [8] tackles a variant of our problem in the process of inferring the topics of interests of online communities starting from the folksonomy they produce. In this version of the task, natural language descriptions are provided for the input keywords, as well as for the target entities, therefore they can be exploited for the alignment. In particular distributional semantic models of words are used to facilitate the mapping of keywords to entities based on their descriptions. Similarly, Nooralahzadeh et al. [12] rely on the knowledge graph of the target resource to named entity recognition and linking.

Whether we consider keyword-based querying, semantic labeling, ontology matching, word sense disambiguation, or related tasks, the key difference with general term mapping is that the former problems are more specific with respect to the resources involved, and more task-oriented than the latter.

## 3 Mapping Terms to DBpedia

We propose a series of approaches to map a set of terms to DBpedia[5], a large knowledge base obtained by automatically parsing semi-structured sections of pages of Wikipedia[6]. While our problem's formulation is agnostic with respect to the knowledge base target of the mapping, some of the features of DBpedia

[5] https://dbpedia.org/
[6] https://wikipedia.org/

enable us to experiment with the methods described in this section. Moreover, DBpedia is a very large and open domain knowledge base, and due to its high connectivity rate to other resources, its position in the Linked Data cloud[7] is essentially that of a central hub.

### 3.1 DBpedia lookup.

The DBpedia project provides a lookup service for keywords.[8] Querying the REST Webservice with a keyword, it returns a list of candidate entities ordered by *refCount*, a measure of the commonness of the entity based on the number of inward links to the resource page in Wikipedia. The candidates are selected by matching the input keyword with the label of a resource, or with an anchor text that was frequently used in Wikipedia to refer to a specific resource. For term mapping, we query the DBpedia lookup service with each input term separately (normalizing the keywords by removing affixes, and replacing underscores with whitespaces) and retrieve the URI of the first result in the response, that is, the resource with the highest refCount.

### 3.2 String Match.

We implemented an alternative algorithm based on string matching. For each input term, if an entity having a matching label (with corrected capitalization) is returned from the DBpedia API, then this entity is returned, otherwise, no label is returned. For instance, given the input keyword `hay_bale`, we perform a HTTP request to the URL http://dbpedia.org/data/Hay_bale and check if the resource exists in DBpedia.

### 3.3 String Match+Redirect.

We also report the performance of the string matching method with the added feature of following the redirection links provided by the resource. For instance, "steel chair" matches dbr:Steel_chair, which in turn redirects to dbr:Folding_chair. The redirection mostly (but not exclusively) helps in cases where there is lexical morphological variation such as plural forms, e.g., dbr:Eggs redirects to dbr:Egg.

### 3.4 Babelfy.

Finally, we tested the performance of a state-of-the-art algorithm for word sense disambiguation and entity linking, Babelfy [10]. Given an input text, Babelfy extracts the spans of text which are most likely to be entities and concepts. For each of these fragments (single words or multi-word expressions), Babelfy generates a list of possible entries according to the semantic network of BabelNet [11].

[7] https://lod-cloud.net/
[8] http://wiki.dbpedia.org/projects/dbpedia-lookup

We query the Babelfy service using all the terms together separated by commas. Partial matches, e.g., https://en.wikipedia.org/wiki/Clothing and https://en.wikipedia.org/wiki/Horse for the keyword `clothes_horse` are discarded.

### 3.5 Modeling the keywords' context with vectors

By analyzing the output of the systems described previously on a pilot test, we identified two main venues for improvement. Firstly, we noted that the vast majority of the missed terms are composed by more than one word, therefore we decided to implement an algorithm that explicitly tackles this issue by splitting the multi-word expressions and searching for entities in DBpedia based on the single words.

The second improvement comes from the observation of the main flaw of the string match method, that is, the disambiguation of each keyword in isolation. As stated in the task definition, this kind of term mapping differs from standard word sense disambiguation because of the lack of context to inform the disambiguation process. However, we can make an assumption on the set of keywords being their own context, i.e., the disambiguation of one keyword provides useful information to disambiguate the other keywords. We therefore test this assumption by encoding it into our novel method for term mapping.

The algorithm, henceforth called Vector-based Contextual disambiguation (VCD) works on top of the string match method, that is, we first run the string match-based algorithm (including the redirection) and save its output, and then run the new algorithm only on the terms that have not been linked in the first step. Formally, the first step consists of running the string match method described in Section 3.2 on the input set of terms $K$, and extracting the set of linked entities $L = SM(K)$. Each term for which an entity is not found in this step is split into the words that compose it, e.g., `basket_of_fruit` $\rightarrow$ [`basket`, `of`, `fruit`]:

$$W = w_1, ..., w_n = split(k_i)$$

For each word, a new entity is retrieved with the string match method as before, e.g., [`basket`, `of`, `fruit`] $\rightarrow$ [`dbr:Basket`, `dbr:Fruit`]:

$$E = e_1, ..., e_m = SM(w_i) \text{ for } w_i \in W \text{ if } SM(w_i) \neq nil$$

Note that the number of retrieved entities ($m$) could be lower than the number of words ($n$), for instance in this example there could be no entity for the word `of`. For each of the new entities, their *semantic similarity* is computed with all the entities that have been previously recognized, and the average is taken[9]:

$$agg\_sim_{AVG}(e_j, L) = \frac{1}{|L|} \sum_{l \in L} sim(e_j, l)$$

[9] We also test a variant where the maximum similarity (MAX) is computed instead of the average similarity (AVG).

This step of the VCD algorithm presupposes a way of computing the semantic similarity between pairs of entities in the target resource. Depending on the target resource, such measure could be already defined, or it could be computed based on lexical and structural properties of the knowledge base. For DBpedia, we rely on the vector space model NASARI [3], in order to obtain pairwise semantic similarity from vectorial representations of the concepts in DBpedia. NASARI is an attempt at building a vector space of concepts starting from the word sense disambiguation of natural language text. The NASARI approach collects cooccurrence information of concepts from Wikipedia and then applies a cluster-based dimensionality reduction. The context of a concept is based on the set of Wikipedia pages where a mention of it is found. The final result is a vector space of 4.5 million 300-dimension dense vectors associated to BabelNet synsets and DBpedia resources. Given two DBpedia resources, we can compute their semantic similarity as the cosine similarity between their corresponding vectors in NASARI.

Finally, the entity $e_j$ with the highest aggregate similarity with the set $L$ of previously disambiguated entities is selected as the disambiguation of the original term. Optionally, a threshold (T) can be imposed on the aggregate similarity score, to avoid linking to entities for which even the highest similarity with the set is still low. This allows us to control the balance between precision and recall, with a lower threshold producing an output for more keywords at the cost of a lower average precision.

## 4 Evaluation

As a use case to experiment with the proposed task, we identify the problem of linking a large database of labeled images to DBpedia. Images come from work in computer vision, and the labels describe relations between objects and locations. Extracting information of this kind is the goal of recent work in information extraction and knowledge base building [2], where authors create a resource of objects and their typical locations by extracting common knowledge from text.

In this section, we describe the large-scale resource for computer vision on which our use case is based, the gold standard dataset we built, and the results we obtained comparing the performances the methods described in Section 3 on the proposed use case.

### 4.1 Data

The SUN database [15] is a large-scale resource for computer vision and objects recognition in images. It comprises 131.067 single images, each of them annotated with a label for the type of scene, e.g., `bedroom` or `dining_room`, and labels for each object identified in the scene, e.g., `wall` or `bed`. The key pieces of information that make the SUN database valuable for applications in robotics and AI is the set of concepts categorized as objects, the set of concepts categorized as scenes, and the implicit relation *locatedIn* between object-scene pairs. However, the concepts in the SUN database are expressed as arbitrary labels, isolated from Linked Data. Figure 1 shows a screenshot from the SUN database Web interface, displaying information relative to an objects and its related scenes in the database. The images have been annotated with 908 categories based on the type of scene (bedroom, garden, airway, ...). Moreover, 313.884 objects were recognized and annotated with one out of 4.479 category labels.

Despite the great amount of work that went into the creation of the SUN database, its applicability to fields related to, but distinct from, computer vision, is hindered by the fact that the set of labels is specific to the resource. To be fair, the creators used the dictionary of lemmas from WordNet [9] to choose the single- or multi-word labels to annotate scenes and objects. However, the labels themselves are not disambiguated, thus they are not directly mapped to any existing resource to promote interoperability.

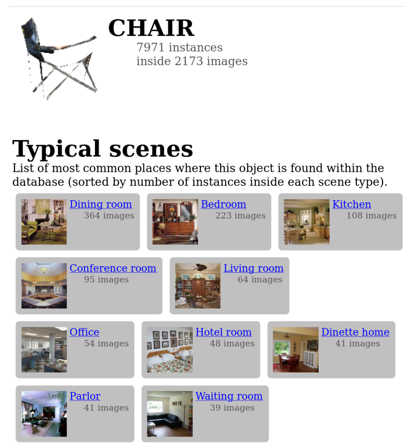


**Fig. 1.** Portion of the SUN database Web interface showing an object (chair) followed by the most frequent scenes associated to it and a set of segmented instances

### 4.2 Gold Standard

In order to assess the difficulty of the mapping the SUN labels to DBpedia, and test the performance of different solutions, a collection of ground truth facts is needed, that is, a set of terms correctly linked to the knowledge base. This set will form the gold standard against which baseline methods and further refined approaches will be tested.

We employed the popular crowdsourcing platform *Figure Eight*[10] to ask paid contributors to manually annotate the object and scene labels from the SUN database. The labels are lowercase English words separated by underscores, e.g., `stand_clothes`, `deer`, `high-heeled_shoes`. About 49% of the labels in both sets are multi-word expressions. The scene labels are prefixed by the starting letter (presumably for the organization of the dataset) and may contain an optional specification after a slash, e.g., `n/newsstand/outdoor` or `b/bakery/shop`.
The task we designed is that of associating a valid URL from the English Wikipedia[11] to each SUN label. This process involved looking up Wikipedia, searching the Web for content related to the keyword, and making non-trivial assignments, to ultimately pair the labels with DBpedia entities. For simple instances, linking a terms to a DBpedia URI is as easy as checking the page with

[10] http://www.figure-eight.com
[11] http://en.wikipedia.org

matching label or a closely related one, or following a redirection, for instance in the cases of differences in spelling such as British vs. American English. A number of cases, however, are not trivial, mostly due to specific concepts being absent from the target knowledge base. To overcome these difficulties that are inherent to the task, we provided a detailed set of guidelines and tips to the annotators, depicted in Figure 2.

**Please use strictly URLs from the English Wikipedia**: https://en.wikipedia.org/wiki/...
Search the page on your favorite search engine, limiting the search result to Wikipedia EN. For instance, on Google you can use a search string like "site:en.wikipedia.org KEYWORD".
You can also search Wikipedia directly using the search box at the top of https://en.wikipedia.org/wiki/Main_Page.
Some keywords will be trivial to match with a Wikipedia page, while others will be more difficult, for instance because a page that matches the keyword exactly does not exist in Wikipedia. The following guidelines will help the task in such cases:

- If a matching Wikipedia page cannot be found, try looking for a slightly
- When facing a choice, try linking to a page that describe the **same kind of concept** as the keyword, e.g. parade_ground → pavement rather than parade_ground → parade.
- **Avoid linking to specific individuals**, such as names of people, cities, ...
- Always **look for alternatives**, even if the keyword has a directly corresponding Wikipedia page.
- There could be misspelled keywords, orthographical variations (e.g., plurals) or different spellings (e.g. British vs. American English). In such cases, **normalize** mentally the keyword and consider the singular form.
- **Avoid Wikipedia disambiguation** pages.

**Fig. 2.** Guidelines for the manual annotation of the keyword linking old standard dataset.

We collected 14,071 single judgments, with at least three independent judgments for each term, from 850 contributors. The entire experiment cost $199 and took roughly one day.

Figure Eight takes automatically care of the aggregation of the contributors' answers, and reports a confidence score associated to each aggregated answer. The confidence score is a measure of the agreement between the annotators on a particular keyword, weighted by a *trust* score assigned to them by Figure Eight based on how they fare in their tasks. The confidence score is of great importance in the analysis of the produced dataset, as we expect the most difficult cases to be associated with a lower confidence score.

Inspecting the resulting data set, we found several classes of difficult cases: from cases where a term can be linked to different, closely related entities (e.g., wood_railing to either dbr:Fence or dbr:Deck_railing[12]) to cases where the input term is complex and not directly represented in the target knowledge base (e.g., basket_of_banana linked to either dbr:Basket or dbr:Banana). There are also errors (mostly spelling mistakes) in the original SUN labels, such as ston→stone or pilow→pillow, that we corrected manually.

We performed a post-processing step to ensure a high-quality gold standard dataset, which led to the removal of 456 entries (9.7% of the total number of original terms) from the gold standard dataset, e.g., because they were links to DBpedia disambiguation pages. The final dataset consists of 4,239 term-DBpedia URI pairs, 3,399 of which are objects and 840 scenes, and it is available at https://project.inria.fr/aloof/data/.

### 4.3 Results

We evaluated the methods introduced in Section 3 against the gold standard sets. This quantitative evaluation consists in counting the number of correctly predicted entity labels and the number of items for which any label is produced at all. With these two pieces of information we can compute the standard metrics used in information retrieval, that is, $precision = \frac{\#correct}{\#retrieved}$ and $recall = \frac{\#correct}{\#instances}$. We also compute the F-score, i.e., the harmonic mean of precision and recall, which summarizes the performance of the methods in a single number: $F\text{-}score = \frac{2 \cdot precision \cdot recall}{precision + recall}$.

The results are shown in Table 1. For VCD, we evaluated the algorithm using as parameters both aggregation methods (AVG and MAX) and a range of thresholds T, and we report only the result of the best combination of parameters (AVG and T=0.3 for the objects, AVG and T=0.8 for the scenes) since the variation of scores was minimal.

**Table 1.** Results of the term mapping systems.

| | Objects | | | Scenes | | |
|---|---|---|---|---|---|---|
| **Method** | **Precision** | **Recall** | **F-score** | **Precision** | **Recall** | **F-score** |
| DBpedia Lookup | 0.397 | 0.272 | 0.322 | 0.449 | 0.412 | 0.430 |
| String Match | 0.523 | 0.327 | 0.402 | 0.598 | 0.535 | 0.564 |
| Babelfy | 0.780 | 0.418 | 0.544 | 0.754 | 0.569 | 0.649 |
| String Match + Redirect | **0.861** | 0.538 | 0.662 | **0.839** | 0.750 | 0.792 |
| VCD | 0.736 | **0.650** | **0.691** | 0.811 | **0.786** | **0.798** |

From the result of the experimentation, it is evident that simple string matching has limited prediction power. Direct application of an entity linking algorithm (Babelfy) also somewhat fails, arguably because of the lack of linguistic context to help the disambiguation of the terms. Wikipedia redirection links, on

[12] In the remainder of the paper, we replace http://dbpedia/org/resource/ with the namespace prefix dbr: to improve readability.

the other hand, represent a powerful mechanism to exploit in order to bridge lexical variations of keywords to the entity labels. It must be noted, though, that Wikipedia redirection is very specific to the resource we chose for our evaluation. In the general case, such useful tool cannot be taken for granted.

The string match method that makes use of redirection links has the best performance in terms of precision both for objects and scenes. This is due to the restricted number of terms that the method is able to retrieve, focusing only on the input entries where there is a perfect string matching and disregarding the ones where the string relation between the entity and the URI is less evident.

VCD obtains the best performance overall, in particular because of its higher recall. The result is particularly notable on the object label dataset.

## 5 Discussion

In addition to the quantitative evaluation presented in the previuos section, we inspected a sample of the output of each method in order to assess their performance qualitatively.

For the DBpedia Lookup baseline, about half of the wrongly predicted labels belong to a named entity (e.g., `oyster_bank` is mapped to dbr:Duchy_of_Cornwall). This behavior is specific of the DBpedia Lookup, indicating a bias towards named entities that makes sense considering the encyclopedic type of resource it targets.

The string match baseline makes less mistakes than the DBpedia lookup, but it still has low precision. Among the terms misclassified by such method, roughly one third are due to the term being in the plural form, or other spelling variations (e.g., `Post-it_note` vs. `Post_it_note`). All these cases, among the others, are corrected by the version of the baseline algorithm that follows the redirects, which obtains a much higher precision score (the highest in the experiment, in fact).

Analyzing the errors committed by the strongest baseline and by Babelfy, we noticed that only a small subset of labels is wrongly predicted by both system, excluding the cases where nothing is returned by one of the systems. Similar figures are found with other pairwise comparisons of the methods. This makes us speculate that a joint system that combines the strengths of several approaches could achieve a much higher performance than any of the single systems.

Providing an additional method for dealing with the cases where the baseline method does not return any entry led to a significant improvement in terms of coverage. As a consequence of considering a wider set of entries, the number of errors increased. The overall performance, however, is better on both datasets.

The novel method we propose, VCD, is designed to solve some of the issues of the general term mapping task, namely the disambiguation of multi-word expressions and the necessity of inferring a notion of context from the input set of keywords. While the experimental results show that solving these problems leads to a better mapping, there are other issues that are not accounted for by any of the presented method. For instance, the *non-compositionality* of multi-word terms is never considered, while the best-performing method (VCD) intrinsically assumes the strict compositionality of the term contituents. For instance, `billiard_ball` must be either a `dbr:Ball` (correct) or a `dbr:Billiard` (wrong), but cannot be linked to any other concept in DBpedia by this algorithm.

Finally, there is an underlying assumption that every keyword in the input set has a corresponding "perfect match" resource in the target resource. In practice, this is hardly the case, and the resulting mismatch calls for slight adaptations of the task definition.

## 6 Conclusion and Future Work

In this paper, we gave a definition of a general term mapping task, aimed at mapping arbitrary sets of terms to a Web Knowledge base, also in relation to well-known tasks in related areas. We built an evaluation framework that includes a manually annotated gold standard dataset and quantitative metrics of performance. In this environment, we tested baseline algorithms based on DBpedia and Wikipedia, and a state-of-the-art system for entity linking, showing their limitations when applied to general term mapping. We then proposed a new approach that fills some of the performance gaps of out-of-the-box solutions, and discussed the results of our experiments, concluding that the most promising ways to approach context-less term mapping are methods that aim for high coverage and employ the whole input set at once to provide context for the term disambiguation.

In the future, besides investigating methods to improve the coverage of the systems, we need to look for general methods that go beyond the specificity of DBpedia, possibly adapting solutions to related problems like the tasks listed in Section 2. We also plan to investigate methods to leverage the asymmetry of the term mapping task as we defined it, that is, solutions that exploits the linguistic features on one side of the mapping and structural features on the opposite side.

## 7 Acknowledgments

* The work of Valerio Basile is partially funded by Progetto di Ateneo/CSP 2016 (*Immigrants, Hate and Prejudice in Social Media*, S1618_L2_BOSC_01).

# Bibliography


[1] Augenstein, I., Gentile, A.L., Norton, B., Zhang, Z., Ciravegna, F.: Mapping Keywords to Linked Data Resources for Automatic Query Expansion, pp. 101–112. Springer Berlin Heidelberg, Berlin, Heidelberg (2013), http://dx.doi.org/10.1007/978-3-642-41242-4_9

[2] Basile, V., Jebbara, S., Cabrio, E., Cimiano, P.: Populating a knowledge base with object-location relations using distributional semantics. In: Knowledge Engineering and Knowledge Management: 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings 20. pp. 34–50. Springer (2016)

[3] Camacho-Collados, J., Pilehvar, M.T., Navigli, R.: Nasari: a novel approach to a semantically-aware representation of items. In: Mihalcea, R., Chai, J.Y., Sarkar, A. (eds.) HLT-NAACL (2015). pp. 567–577. ACL (2015), http://dblp.uni-trier.de/db/conf/naacl/naacl2015.html

[4] Damme, C.V., Hepp, M., Siorpaes, K.: Folksontology: An integrated approach for turning folksonomies into ontologies. In: Bridging the Gep between Semantic Web and Web 2.0 (SemNet 2007). pp. 57–70 (2007), http://www.kde.cs.uni-kassel.de/ws/eswc2007/proc/FolksOntology.pdf

[5] Freitas, A., Curry, E., Oliveira, J.G., O'Riain, S.: Querying heterogeneous datasets on the linked data web: Challenges, approaches, and trends. IEEE Internet Computing 16(1), 24–33 (2012), http://dx.doi.org/10.1109/MIC.2011.141

[6] Kaiser, P., Lewis, M., Petrick, R.P.A., Asfour, T., Steedman, M.: Extracting common sense knowledge from text for robot planning. In: 2014 IEEE International Conference on Robotics and Automation, ICRA 2014, Hong Kong, China, May 31 - June 7, 2014. pp. 3749–3756 (2014), http://dx.doi.org/10.1109/ICRA.2014.6907402

[7] Limpens, F.: Multi-points of view semantic enrichment of folksonomies. Theses, Université Nice Sophia Antipolis (Oct 2010), https://tel.archives-ouvertes.fr/tel-00530714

[8] Meng, Z.: Temporal and semantic analysis of richly typed social networks from user-generated content sites on the Web. Theses, Université Nice Sophia Antipolis [UNS] (Nov 2016), https://hal.inria.fr/tel-01402612

[9] Miller, G.A.: Wordnet: A lexical database for english. Commun. ACM 38(11), 39–41 (Nov 1995), http://doi.acm.org/10.1145/219717.219748

[10] Moro, A., Raganato, A., Navigli, R.: Entity Linking meets Word Sense Disambiguation: a Unified Approach. Transactions of the Association for Computational Linguistics (TACL) 2, 231–244 (2014)

[11] Navigli, R., Ponzetto, S.P.: BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial Intelligence 193, 217–250 (2012)

[12] Nooralahzadeh, F., Lopez, C., Cabrio, E., Gandon, F., Segond, F.: Adapting Semantic Spreading Activation to Entity Linking in text. In: Proceedings

of NLDB 2016 - 21st International Conference on Applications of Natural Language to Information Systems. Manchester, United Kingdom (Jun 2016), https://hal.inria.fr/hal-01332626

[13] Specia, L., Motta, E.: Integrating folksonomies with the semantic web. 4th European Semantic Web Conference (2007), http://www.eswc2007.org/

[14] Tenorth, M., Beetz, M.: KnowRob – Knowledge Processing for Autonomous Personal Robots. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 4261–4266 (2009)

[15] Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: CVPR. pp. 3485–3492. IEEE Computer Society (2010), http://dblp.uni-trier.de/db/conf/cvpr/cvpr2010.html#XiaoHEOT10

[16] Young, J., Basile, V., Kunze, L., Cabrio, E., Hawes, N.: Towards Lifelong Object Learning by Integrating Situated Robot Perception and Semantic Web Mining. In: Proceedings of the European Conference on Artificial Intelligence (ECAI) 2016 conference. THe Hague, Netherlands (Aug 2016), https://hal.inria.fr/hal-01370140