Populating a Knowledge Base with Object-Location Relations Using Distributional Semantics

Valerio Basile, Soufian Jebbara, Elena Cabrio, Philipp Cimiano

Populating a Knowledge Base with Object-Location Relations using Distributional Semantics

Valerio Basile[1], Soufian Jebbara[2], Elena Cabrio[1], and Philipp Cimiano[2]

[1] Université Côte d'Azur, Inria, CNRS, I3S, France, email: valerio.basile@inria.fr; elena.cabrio@unice.fr

[2] Bielefeld University, Germany, email: {sjebbara,cimiano}@cit-ec.uni-bielefeld.de

Abstract. The paper presents an approach to extract knowledge from large text corpora, in particular knowledge that facilitates object manipulation by embodied intelligent systems that need to act in the world. As a first step, our goal is to extract the prototypical location of given objects from text corpora. We approach this task by calculating relatedness scores for objects and locations using techniques from distributional semantics. We empirically compare different methods for representing locations and objects as vectors in some geometric space, and we evaluate them with respect to a crowd-sourced gold standard in which human subjects had to rate the prototypicality of a location given an object. By applying the proposed framework on DBpedia, we are able to build a knowledge base of 931 high confidence object-locations relations in a fully automatic fashion.[3]

1 Introduction

Embodied intelligent systems such as robots require world knowledge to be able to perceive the world appropriately and perform appropriate actions on the basis of their understanding of the world. Take the example of a domestic robot that has the task of tidying up an apartment. A robot needs, e.g., to categorize different objects in the apartment, know where to put or store them, know where and how to grasp them, and so on. Encoding such knowledge by hand is a tedious, time-consuming task and is inherently prone to yield incomplete knowledge. It would be desirable to develop approaches that can extract such knowledge automatically from data.

To this aim, in this paper we present an approach to extract object knowledge from large text corpora. Our work is related to the machine reading and open information extraction paradigms aiming at learning generic knowledge from text corpora. In contrast, in our research we are interested in particular in extracting knowledge that facilitates object manipulation by embodied intelligent systems that need to act in the world. Specifically, our work focuses on the problem of *relation extraction* between entities mentioned in the text[4]. A *relation* is defined in the form of a tuple $t = (e_1; e_2; ...; e_n)$ where the e_i are entities in a predefined relation r within document D [1]. We develop a framework with foundations in *distributional semantics*, the area of Natural Language

[3] The work in this paper is partially funded by the ALOOF project (CHIST-ERA program).

[4] In the rest of the paper, the labels of the entities are identifiers from DBpedia URIs, stripped of the namespace `http://dbpedia.org/resource/` for readability.

Processing that deals with the representation of the meaning of words in terms of their distributional properties, i.e., the context in which they are observed. It has been shown in the literature that distributional semantic techniques give a good estimation of the *relatedness* of concepts expressed in natural language (see Section 3 for a brief overview of distributional semantics principles). Semantic relatedness is useful for a number of tasks, from query expansion to word association, but it is arguably too general to build a general knowledge base, i.e., a triple like `<entity1, relatedTo, entity2>` might not be informative enough for many purposes.

Distributional Relation Hypothesis We postulate that the relatedness relation encoded in distributional vector representations can be made more precise based on the type of the entities involved in the relation, i.e., if two entities are distributionally related, the natural relation that comes from their respective types is highly likely to occur. For example, the *location* relation that holds between an *object* and a *room* is represented in a distributional space if the entities representing the object and the room are highly associated according to the distributional space's metric.

Based on this assumption, as a first step of our work, we extract the prototypical location of given objects from text corpora. We frame this problem as a ranking task in which, given an object, our method computes a ranking of locations according how protoypical a location they are for this object. We build on the principle of distributional similarity and map each location and object to a vector representation computed on the basis of words these objects or locations co-occur with in a corpus. For each object, the locations are then ranked by the cosine similarity of their vector representations.

The paper is structured as follows. Section 2 discusses relevant literature, while Section 3 provides a background on word and entity vector spaces. Section 4 describes the proposed framework to extract relations from text. Section 5 reports on the creation of the goldstandard, and on the experimental results. Section 6 describes the obtained knowledge base of object locations, while conclusions end the paper.

2 Related Work

Our work relates to the three research lines discussed below, i.e.: *i)* machine reading, *ii)* supervised relation extraction, and *iii)* encoding common sense knowledge in domain-independent ontologies and knowledge bases.

The machine reading paradigm. In the field of knowledge acquisition from the Web, there has been substantial work on extracting taxonomic (e.g. hypernym), part-of relations [17] and complete qualia structures describing an object [9]. Quite recently, there has been a focus on the development of systems that can extract knowledge from any text on any domain (the open information extraction paradigm [15]). The DARPA Machine Reading Program [2] aims at endowing machines with capabilities for lifelong learning by automatically reading and understanding texts (e.g. [14]). While such approaches are able to quite robustly acquire knowledge from texts, these models are not sufficient to meet our objectives since: *i)* they lack visual and sensor-motor grounding,

ii) they do not contain extensive object knowledge. Thus, we need to develop additional approaches that can harvest the Web to learn about usages, appearance and functionality of common objects. While there has been some work on grounding symbolic knowledge in language [32], so far there has been no serious effort to compile a large and grounded object knowledge base that can support cognitive systems in understanding objects.

Supervised Relation Extraction. While machine reading attempts to acquire general knowledge by reading texts, other works attempt to extract specific relations applying supervised techniques to train classifiers. A training corpus in which the relation of interest is annotated is typically assumed (e.g. [7]). Another possibility is to rely on the so called *distant supervision* assumption and use an existing knowledge base to bootstrap the process by relying on triples or facts in the knowledge base to label examples in a corpus (e.g. [43, 21, 20, 40]). Other researchers have attempted to extract relations by reading the Web, e.g. [5]. Our work differs from these approaches in that, while we are extracting a specific relation, we do not rely on supervised techniques to train a classification model, but rather rely on semantic relatedness and distributional similarity techniques to populate a knowledge base with the relation in question.

Ontologies and KB of common sense knowledge. DBpedia[5] is a large-scale knowledge base automatically extracted from semi-structured parts of Wikipedia. Besides its sheer size, it is attractive for the purpose of collecting general knowledge given the one-to-one mapping with Wikipedia (allowing us to exploit the textual and structural information contained in there) and its position as the central hub of the Linked Open Data cloud.

YAGO [38] is an ontology automatically created by mapping relations between WordNet synsets such as hypernymy and relations between Wikipedia pages such as links and redirects to semantic relations between concepts. Despite its high coverage, for our goals YAGO suffers from the same drawbacks of DBpedia, i.e. a lack of general relations between entities that are not instance of the DBpedia ontology, such as common objects. While a great deal of relations and properties of named entities are present, knowledge about, e.g. the location or the functionality of entities is missing.

ConceptNet[6] [26] is a semantic network containing lots of things computers should know about the world. While it shares the same goals of the knowledge base we aim at building, ConceptNet is not a Linked Open Data resource. In fairness, the resource is in a graph-like structure, thus RDF triples could be extracted from it, and the building process provides a way of linking the nodes to DBpedia entities, among other LOD resources. However, we cannot integrate ConceptNet directly in our pipeline because of the low coverage of the mapping with DBpedia— of the 120 DBpedia entities in our gold standard (see Section 5) only 23 have a correspondent node in ConceptNet.

OpenCyC[7] attempts to assemble a comprehensive ontology and knowledge base of everyday common sense knowledge, with the goal of enabling AI applications to perform human-like reasoning. While for the moment in our work we focus on specific

[5] `http://dbpedia.org`
[6] `http://conceptnet5.media.mit.edu/`
[7] `http://www.opencyc.org/`; as RDF representations: `http://sw.opencyc.org/`

concepts and relations relevant to our scenario, we will consider linking them to real-world concepts in OpenCyc.

3 Background: Word and Entity Vector Spaces

Word space models (or distributional space models, or word vector spaces) are abstract representations of the meaning of words, encoded as vectors in a high-dimensional space. A word vector space is constructed by counting *cooccurrences* of pairs of words in a text corpus, building a large square n-by-n matrix where n is the size of the vocabulary and the cell i, j contains the number of times the word i has been observed in cooccurrence with the word j. The i-th row in a cooccurrence matrix is a n-dimensional vector that acts as a *distributional* representation of the i-th word in the vocabulary. Words that appear in similar contexts often have similar representations in the vector space; this similarity is geometrically measurable with a distance metric such as cosine similarity, defined as the cosine of the angle between two vectors. This is the key point to linking the vector representation to the idea of semantic relatedness, as the *distributional hypothesis* states that "words that occur in the same contexts tend to have similar meaning" [19]. Several techniques can be applied to reduce the dimensionality of the cooccurrence matrix. Latent Semantic Analysis [24], for instance, uses Singular Value Decomposition to prune the less informative elements while preserving most of the topology of the vector space, and reducing the number of dimensions to 100-500.

In parallel, neural network-based models have recently began to rise to prominence. To compute word embeddings, several models rely on huge amounts of natural language texts from which a vector representation for each word is learned by a neural network. Their representations of the words are based on *prediction* as opposed to *counting* [4].

Vector spaces created on word distributional representations have been successfully proven to encode word similarity and relatedness relations [34, 36, 10], while word embeddings have proven to be a useful feature in many natural language processing tasks [12, 25, 37] in that they often encode semantically meaningful information of a word.

4 Word Embeddings for Relation Extraction

This section presents our framework to extract relations from natural language text. The methods are based on distributional semantics, but present different approaches to compute vector representations of entities: one is based on a word embedding approach (Section 4.1), the other on a LSA-based representation of DBpedia entities (Section 4.2). We present one framework for which we test different ways of calculating the vector embeddings, each one having its own specificities and strengths.

4.1 A Word Space Model of Entity Lexicalizations

In this section, we propose a neural network-based word embedding method for the automatic population of a knowledge base of object-location relations. As outlined in Section 1, we frame this task as a ranking problem and score the vector representation

for object-location pairs with respect to how prototypical the location is for the given object. Many word embedding methods encode useful semantic and syntactic properties [23, 31, 29] that we leverage for the extraction of object-location relations. In this work, we restrict our experiments to the skip-gram method [28]. The objective of the skip-gram method is to learn word representations that are useful for predicting context words. As a result, the learned embeddings often display a desirable linear structure [31, 29]. In particular, word representations of the skip-gram model often produce meaningful results using simple vector addition [29]. For this work, we trained the skip-gram model on a corpus of roughly 83 million Amazon reviews [27].

Motivated by the compositionality of word vectors, we derive vector representations for the entities as follows: considering a DBpedia entity such as `Public_Toilet` (we call this label the *lexicalization*), we clean it by removing parts in parenthesis, convert it to lower case, and split it into its individual words. We retrieve the respective word vectors from our pretrained word embeddings and sum them to obtain a single vector, namely, the vector representation of the entity: $vector(public\_toilet) = vector(public) + vector(toilet)$. The generation of entity vectors is trivial for "single-word" entities, such as `Cutlery` or `Kitchen`, that are already contained in our word vector vocabulary. In this case, the entity vector is simply the corresponding word vector. With this derived set of entity vector representations, we compute cosine vector similarity score for object-location pairs. This score is an indicator of how typical the location for the object is. Given an object, we can create a ranking of locations with the most likely location candidates at the top of the list (see Table 1).

Table 1: Locations for a sample object, extracted by computing cosine similarity on skip-gram-based vectors.

Object	**Location**	**Cosine Similarity**
Dishwasher	Kitchen	.636
	Laundry_room	.531
	Pantry	.525
	Wine_cellar	.519

4.2 Distributional Representations of Entities

Vector representations of words (Section 4.1) are attractive since they only require a sufficiently large text corpus with no manual annotation. However, the drawback of focusing on words is that a series of linguistic phenomena may affect the vector representation. For instance, a polysemous word as *rock* (stone, musical genre, metaphorically strong person, etc.) is represented by a single vector where all the senses are conflated.

NASARI [8], a resource containing vector representations of most of DBpedia entities, solves this problem by building a vector space of concepts. The NASARI vectors are actually distributional representations of the entities in BabelNet [33], a large multilingual lexical resource linked to Wordnet, DBpedia, Wiktionary and other resources. The NASARI approach collects cooccurrence information of concepts from Wikipedia and then applies a LSA-like procedure for dimensionality reduction. The context of a

concept is based on the set of Wikipedia pages where a mention of it is found. As shown in [8], the vector representations of entities encode some form of semantic relatedness, with tests on a sense clustering task showing positive results. Table 2 shows a sample of pairs of NASARI vectors together with their pairwise cosine similarity ranging from -1 (totally unrelated) to 1 (identical vectors).

Table 2: Examples of cosine similarity computed on NASARI vectors.

	Cherry	Microsoft
Apple	.917	.325
Apple_Inc.	.475	.778

Following the hypothesis put forward in the introduction, we focus on the extraction of object-location relations by computing the cosine similarities of object and location entities. We exploit the alignment of BabelNet with DBpedia, thus generating a similarity score for pairs of DBpedia entities. For example, the DBpedia entity `Dishwasher` has a cosine similarity of .803 to the entity `Kitchen`, but only .279 with `Classroom`, suggesting that the appropriate location for a generic dishwasher is the kitchen rather than a classroom. Since cosine similarity is a graded value on a scale from -1 to 1, we can generate, for a given object, a ranking of candidate locations, e.g., the rooms of a house. Table 3 shows a sample of object-location pairs of DBpedia labels, ordered by the cosine similarity of their respective vectors in NASARI. Prototypical locations for the objects show up at the top of the list as expected, indicating a relationship between the semantic relatedness expressed by the cosine similarity of vector representations and the actual locative relation of entities.

Table 3: Locations for a sample object, extracted by computing cosine similarity on NASARI vectors.

Object	**Location**	**Cos. Similarity**
Dishwasher	Kitchen	.803
	Air_shower_(room)	.788
	Utility_room	.763
	Bathroom	.758

5 Evaluation

This section presents the evaluation of the proposed framework for relation extraction (Section 4). We collected a set of relations rated by human subjects to provide a common benchmark, and we test several methods with varying values for their parameters. We then adopt the best performing method to automatically build a knowledge base and test its quality against the manually created gold standard dataset.

5.1 Gold Standard

To test our hypothesis, we collected a set of human judgments about the likelihood of objects to be found in certain locations. To select the objects and locations for this experiment, every DBpedia entity that falls under the category `Domestic_implements`, or under one of the `narrower` categories than `Domestic_implements` according to SKOS[8], is considered an object; every DBpedia entity that falls under the category `Rooms` is considered a location. This step results in 336 objects and 199 locations.

To select suitable object-location pairs for the creation of the gold standard, we need to filter out odd or uncommon examples of objects or locations like `Ghodiyu` or `Fainting_room`. For example, the rankings produced by the cosine similarity of NASARI vectors (Table 3) are cluttered with results that are less prototypical because of their uncommonness. An empirical measure of commonness of entities could be used to rerank or filter the result to improve its generality. To this extent, we use the URI counts extracted from the parsing of Wikipedia with the DBpedia Spotlight tool for entity linking [13]. These counts are derived, for each DBpedia entity, from the number of incoming links to its correspondent Wikipedia page. We use it as an approximation of the notion of commonness of locations, e.g., a `Kitchen` (URI count: 742) is a more common location than a `Billiard_room` (URI count: 82). Table 4 shows an example of using such counts to filter out irrelevant entries from the ranked list of candidate locations for the entity `Paper_towel` according to NASARI-based similarity.

Table 4: Locations for `Paper_towel`, extracted by computing cosine similarity on NASARI vectors with URI count. Locations with frequency <100 are in gray.

Location	URI count	Cosine Similarity
Air_shower_(room)	0	.671
Public_toilet	373	.634
Mizuya	11	.597
Kitchen	742	.589

We rank the 66,864 pairs of `Domestic_implements` and `Rooms` using the aforementioned entity frequency measure and select the 100 most frequent objects and the 20 most frequent locations (2,000 object-location pairs in total). Examples of pairs: (`Toothbrush,Hall`), (`Wallet, Ballroom`) and (`Nail_file, Kitchen`).

In order to collect the judgments, we set up a crowdsourcing experiment on the Crowdflower platform[9]. For each of the 2,000 object-location pairs, contributors were asked to rate the likelihood of the object to be in the location out of four possible values:

- **-2** (**unexpected**): finding the object in the room would cause surprise, e.g., it is unexpected to find a bathtub in a cafeteria.
- **-1** (**unusual**): finding the object in the room would be odd, the object feels out of place, e.g., it is unusual to find a mug in a garage.

[8] Simple Knowledge Organization System: `https://www.w3.org/2004/02/skos/`
[9] `http://www.crowdflower.com/`

– **1** (**plausible**): finding the object in the room would not cause any surprise, it is seen as a normal occurrence, e.g., it is plausible to find a funnel in a dining room.
– **2** (**usual**): the room is the place where the object is typically found, e.g, the kitchen is the usual place to find a spoon.

Contributors are shown ten examples per page, instructions, a short description of the entities (the first sentence from the Wikipedia abstract), a picture (from Wikimedia Commons, when available), and the list of possible answers as labeled radio buttons.

After running the crowdsourcing experiment for a few hours, we collected 12,767 valid judgments (455 were deemed "untrusted" by Crowdflower's quality filtering system based on a number of test questions we provided). Most of the pairs have received at least 5 separate judgments, with some outliers collecting more than one hundred judgments each. The average agreement, i.e. percentage of contributors that answered the most common answer for a given question, is 64.74%. The judgments are skewed towards the negative end of the spectrum, as expected, with 37% pairs rated unexpected, 30% unusual, 24% plausible and 9% usual. The cost of the experiment was 86 USD.

5.2 Ranking Evaluation

The proposed methods produce a ranking on top of a list of locations, given an input object. To test the validity of our methods we need to compare their output against a gold standard ranking. The latter is extracted from the dataset described in Section 5.1 by assigning to each object-location pair the average of the numeric values of the judgments received. For instance, if the pair (`Wallet, Ballroom`) has been rated -2 (unexpected) six times, -1 (unusual) three times, and never 1 (plausible) or 2 (usual), its score will be about -1.6, indicating that a `Wallet` is not very likely to be found in a `Ballroom`. The pairs are then ranked by this averaged score on a per-object basis.

As a baseline, we apply two simple methods based on entity frequency. In the *location frequency* baseline, the object-location pairs are ranked according to the frequency of the location. The ranking is thus the same for each object, since the score of a pair is only computed based on the location. This method makes sense in absence of any further information on the object: e,g, a robot tasked to find an unknown object should inspect "common" rooms such as a kitchen or a studio first, rather than "uncommon" rooms such as a pantry. The second baseline (*link frequency*) is based on counting how often every object is mentioned on the Wikipedia page of every location and vice versa. A ranking is produced based on these counts. An issue is that they could be sparse, i.e., most object-location pairs have a count of 0, thus sometimes producing no value for the ranking for an object. This is the case for rather "unusual" objects and locations.

For each object in the dataset, we compare the location ranking produced by our algorithms to the gold standard ranking and compute the Normalized Discounted Cumulative Gain (NDCG), a measure of rank correlation used in information retrieval that gives more weight to the results at the top of the list than at its bottom. This choice of evaluation metric follows from the idea that it is more important to guess the position in the ranking of most likely locations for a given object than to the least likely locations. Table 5 shows the average NDCG across all objects: methods *NASARI-sim* (Section 4.2) and *SkipGram-sim* (Section 4.1), plus the two baselines introduced above. Both our methods outperform the baselines with respect to the gold standard rankings.

Table 5: Average NDCG of the produced rankings against the gold standard rankings.

Method	NDCG
Location frequency baseline	.851
Link frequency baseline	.875
NASARI-sim	.903
SkipGram-sim	.912

5.3 Precision Evaluation

The NDCG measure gives a complete account of the quality of the produced rankings, but it is not easy to interpret apart from comparisons of different outputs. To gain a better insight into our results, we provide an alternative evaluation based on the "precision at k" measure. This Information Retrieval measure is the number of retrieved items that are ranked in the top-k part of the retrieved list and of the relevance ranking. In our experiments, for a given object, precision at k is the number of locations among the first k of the produced rankings that are also among the top-k locations in the gold standard ranking. It follows that, with $k = 1$, precision at 1 is 1 if the top returned location is the top location in the gold standard, and 0 otherwise. We compute the average of precision at k for $k = 1$ and $k = 3$ across all the objects. The results are shown in Table 6.

Table 6: Average precision at k for $k = 1$ and $k = 3$.

Method	precision at 1	precision at 3
Location frequency baseline	.000	.008
Link frequency baseline	.280	.260
NASARI-sim	.390	.380
SkipGram-sim	.350	.400

As for the rank correlation evaluation, our methods outperform the baselines. The location frequency baseline performs very poorly, due to an idiosyncrasy in the frequency data, that is, the most "frequent" location in the dataset is *Aisle*. This behavior reflects the difficulty in evaluating this task using only automatic metrics, since automatically extracted scores and rankings may not correspond to common sense judgment.

The NASARI-based similarities outperform the SkipGram-based method when it comes to guessing the most likely location for an object, as opposed to the better performance of SkipGram-sim in terms of precision at 3 and rank correlation (Section 5.2).

We explored the results and found that for 19 objects out of 100, NASARI-sim correctly guesses the top ranking location but SkipGram-sim fails, while the opposite happens 15 out of 100 times. We also found that the NASARI-based method has a lower coverage than the other method., due to the coverage of the original resource (NASARI), where not every entity in DBpedia is assigned a vector (objects like `Backpack` and `Comb`, and locations like `Loft` are all missing). The SkipGram-based method also suffer from this problem, however, only for very rare or uncommon objects and locations (as `Triclinium` or `Jamonera`). These findings suggest that the two methods

could have different strengths and weaknesses. In the following section we show two strategies to combine them.

5.4 Hybrid Methods: Fallback Pipeline and Linear Combination

The results from the previous sections highlight that the performance of our two main methods may differ qualitatively. In an effort to overcome the coverage issue of NASARI-sim, and at the same time experiment with hybrid methods to extract location relations, we devised two simple ways of combining the SkipGram-sim and NASARI-sim methods. The first method is based on a fallback strategy: given an object, we consider the pair similarity of the object to the top ranking location according to NASARI-sim as a measure of confidence. If the top ranked location among the NASARI-sim ranking is exceeding a certain threshold, we consider the ranking returned by NASARI-sim as reliable. Otherwise, if the similarity is below the threshold, we deem the result unreliable and we adopt the ranking returned by SkipGram-sim instead. The second method produces an object-location similarity scores by linear combination of the NASARI and SkipGram similarities. The similarity score for the generic pair o, l is thus given by $sim(o, l) = \alpha sim_{NASARI}(o, l) + (1 - \alpha) sim_{SkipGram}(o, l)$, where parameter α controls the weight of one method w.r.t. the other.

Table 7: Rank correlation and precision at k for the method based on fallback strategy.

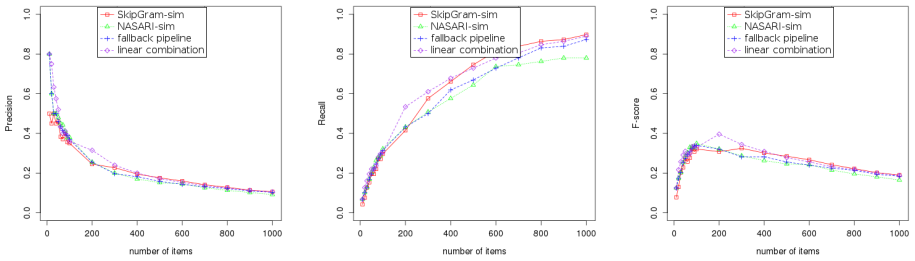
Method	**NDCG**	**precision at 1**	**precision at 3**
Fallback strategy (threshold=.4)	.907	.410	.393
Fallback strategy (threshold=.5)	.906	.400	.393
Fallback strategy (threshold=.6)	.908	.410	.406
Fallback strategy (threshold=.7)	.909	.370	.396
Fallback strategy (threshold=.8)	.911	.360	.403
Linear combination (α=.0)	.912	.350	.400
Linear combination (α=.2)	.911	.380	.407
Linear combination (α=.4)	.913	.400	.423
Linear combination (α=.6)	.911	.390	.417
Linear combination (α=.8)	.910	.390	.410
Linear combination (α=1.0)	.903	.390	.380
Max	.911	.410	.413

Table 7 shows the obtained results, with varying values of the parameters $threshold$ and α. The line labeled *Max* shows the result obtained by choosing the highest similarity between NASARI-sim and SkipGram-sim, for comparison. While the NDCG is basically not affected, both precision at 1 and precision at 3 show an increase in performance with respect to any of the previous methods.

6 Building a Knowledge Base of Object Locations

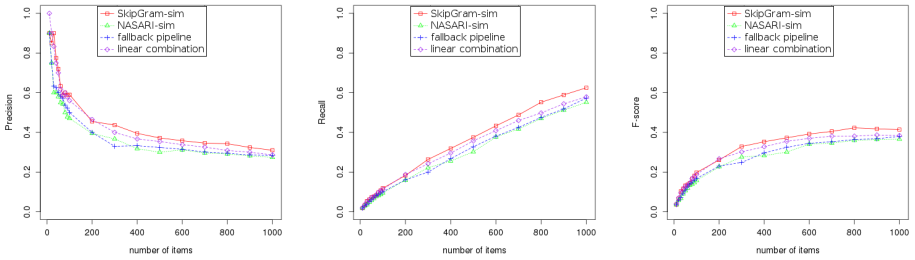
In the the previous section, we tested how the proposed methods succeed in determining the relation between given objects and locations on a closed set of entities (for the

purpose of evaluation). In this section we return to the original motivation of this work, that is, to collect location information about objects in an automatic fashion.



(a) Precision (b) Recall (c) F-score

Fig. 1: Evaluation on automatically created knowledge bases (“usual” locations).



(a) Precision (b) Recall (c) F-score

Fig. 2: Evaluation on automatically created knowledge bases (“plausible” and “usual” locations).

All the methods introduced in this work are based on some measure of relatedness between entities, expressed as a real number in the range [-1,1] interpretable as a sort of confidence score relative to the target relation. Therefore, by imposing a threshold on the similarity scores and selecting only the object-location pairs that score above said threshold, we can extract a high-confidence set of object-location relations to build a new knowledge base from scratch. Moreover, by using different values for the threshold, we are able to control the quality and the coverage of the produced relations.

We test this approach on the gold standard dataset introduced in Section 5, using the version with data aggregated by Crowdflower: the constributors’ answers are aggregated using relative majority, that is, each object-location pair has exactly one judgment

assigned to it, corresponding to the most popular judgment among all the contributors that answered that question.We extract two lists of relations from this dataset to be used as a gold standard for experimental tests: one list of the 156 pairs rated 2 (*usual*) by the majority of contributors, and a larger list of the 496 pairs rated either 1 (*plausible*) or 2 (*usual*). The aggregated judgments in the gold standard have a confidence score assigned to them by Crowdflower, based on a measure of inter-rater agreement. Pairs that score low on this confidence measure ($\leqslant 0.5$) were filtered out, leaving respectively 118 pairs in the "usual" set 496 pairs in the "plausible or usual" set.

We order the object-location pairs produced by our two main methods by similarity score, and select the first n from the list, with n being a parameter. We also add to the comparison the results of the two hybrid methods from Section 5.4, with the best performing parameters in terms of precision at 1, namely the fallback strategy with threshold on similarity equal to 0.6 and the linear combination with $\alpha = 0.4$. For the location relations extracted with these methods, we compute the precision and recall against the gold standard sets, with varying values of n. Here, the precision is the percentage of correctly predicted pairs in the set of all predicted pairs, while the recall is the percentage of predicted pairs that also occur in the gold standard. Figures 1 and 2 show the evaluation of the four methods evaluated against the two aggregated gold standard datasets described above. Figures 1c and 2c, in particular, show F-score plots for a direct comparison of the performance. The precision and recall figures show similar performances for all the methods, with the SkipGram-sim method obtaining a generally higher recall. The SkipGram-sim method produces generally better-quality sets of relations. However, if the goal is high precision, the other methods may be preferable.

Given these results, we can aim for a high-confidence knowledge base by selecting the threshold on object-location similarity scores that produces a reasonably high precision knowledge base in the evaluation. For instance, the knowledge base made by the top 50 object-location pairs extracted with the linear combination method ($\alpha = 0.4$) has 0.52 precision and 0.22 recall on the "usual" gold standard (0.70 and 0.07 respectively on the "usual" or "plausible" set, see Figures 1a and 2a). The similarity scores in this knowledge base range from 0.570 to 0.866. Following the same methodology that we used to construct the gold standard set of objects and locations (Section 5.1), we extract all the 336 `Domestic_implements` and 199 `Rooms` from DBpedia, for a total of 66,864 object-location pairs. Selecting only the pairs whose similarity score is higher than 0.570, according to the linear combination method, yields 931 high confidence location relations. Of these, only 52 were in the gold standard set of pairs (45 were rated "usual" or "plausible" locations), while the remaining 879 are new, such as (`Trivet`, `Kitchen`), (`Flight_bag`, `Airport_lounge`) or (`Soap_dispenser`, `Unisex_public_toilet`). The distribution of objects across locations has an arithmetic mean of 8.9 objects per location and standard deviation 11.0.`Kitchen` is the most represented location with 89 relations, while 15 out of 107 locations are associated with one single object. [10]

[10] The full automatically created knowledge base is available at `http://project.inria.fr/aloof/files/2016/04/objectlocations.nt_.gz`

7 Conclusion and Future Work

This paper presents novel methods to extract object relations, focusing on the typical locations of common objects. The proposed approaches are based on distributional semantics, where vector spaces are built that represent words or concepts in a high-dimensional space. We then map vector distance to semantic relatedness, and instantiate a specific relation that depends on the type of the entities involved (e.g., an object highly related to a room indicates that the room is a typical location for the object)[11].

The NASARI-based scoring method is a concept-level vectors space model derived from BabelNet. The skip-gram model (Section 4.1) is trained on Amazon review data and offers a word-level vector space which we exploit for scoring object-location pairs. Experiments on a crowdsourced dataset of human judgments show that they offer different advantages. To combine their strengths, we test two combination strategies, and show an improvement on their performances. Finally, we select the best parameters to extract a new, high-precision knowledge base of object locations.

As future work, we would like to employ *retrofitting* [16] to enrich our pretrained word embeddings with concept knowledge from a semantic network such as ConceptNet or WordNet [30] in a post-processing step. With this technique, we might be able to combine the benefits of the concept-level and word-level semantics in a more sophisticated way to bootstrap the creation of an object-location knowledge base. We believe that this method is a more appropriate tool than the simple linear combination of scores. By specializing our skip-gram embeddings for relatedness instead of similarity [22] even better results could be achieved. Apart from that, we would like to investigate knowledge base embeddings and graph embeddings [42, 6, 39] that model entities and relations in a vector space in more detail. By defining an appropriate training objective, we might be able to compute embeddings that encode directly object-location relations and thus are tailored more precisely to our task at hand. Finally, we used the frequency of entity mentions in Wikipedia as a measure of commonality to drive the creation of a gold standard set for evaluation. This information, or equivalent measures, could be integrated directly into our relation extraction framework, for example in the form of a weighting scheme, to improve its predictions accuracy.

As main limitation of our current work, it needs to be stressed that the relation in question (here *isLocatedAt*) is predicted in all cases where the semantic relatedness is over a certain threshold. Thus, the method described is not specific for the particular relation given. In fact, the relation we predict is a relation of general (semantic) association. In our particular case, the method works due to the specifiy of the types invovled (room and object), which seem to be specific enough to restict the space of possible relations. It is not clear, however, to which other relations our method would generalize. This is left for future investigation. In particular, we intend to extend our method so that a model can be trained to predict a particular relation rather than a generic associative relationship.

[11] All the datasets resulting from this work are available at `https://project.inria.fr/aloof/data/`

References

1. Bach, N., Badaskar, S.: A Review of Relation Extraction (2007)
2. Barker, K., Agashe, B., Chaw, S.Y., Fan, J., Friedland, N., Glass, M., Hobbs, J., Hovy, E., Israel, D., Kim, D.S., Mulkar-Mehta, R., Patwardhan, S., Porter, B., Tecuci, D., Yeh, P.: Learning by reading: A prototype system, performance baseline and lessons learned. In: Procs of the 22Nd National Conference on Artificial Intelligence - Volume 1. pp. 280–286. AAAI'07 (2007)
3. Baroni, M., Bernardi, R., Zamparelli, R.: Frege in space: A program for compositional distributional semantics. Linguistic Issues in Language Technology 9, 5–110 (2014)
4. Baroni, M., Dinu, G., Kruszewski, G.: Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In: Procs of ACL 2014 (Volume 1: Long Papers) (June 2014)
5. Blohm, S., Cimiano, P., Stemle, E.: Harvesting relations from the web - quantifiying the impact of filtering functions. In: Procs of the Twenty-Second AAAI Conference on Artificial Intelligence. pp. 1316–1321 (2007)
6. Bordes, A., Weston, J., Collobert, R., Bengio, Y.: Learning Structured Embeddings of Knowledge Bases. Artificial Intelligence (Bengio), 301–306
7. Bunescu, R.C., Mooney, R.J.: A shortest path dependency kernel for relation extraction. In: HLT/EMNLP. `http://acl.ldc.upenn.edu/H/H05/H05-1091.pdf`
8. Camacho-Collados, J., Pilehvar, M.T., Navigli, R.: In: HLT-NAACL (2015)
9. Cimiano, P., Wenderoth, J.: Automatically learning qualia structures from the web. In: Procs of the ACL-SIGLEX Workshop on Deep Lexical Acquisition. pp. 28–37. DeepLA '05 (2005)
10. Ciobanu, A.M., Dinu, A.: Alternative measures of word relatedness in distributional semantics. In: Joint Symposium on Semantic Processing. p. 80 (2013)
11. Clark, S., Coecke, B., Sadrzadeh, M.: A compositional distributional model of meaning. In: Proceedings of the Second Quantum Interaction Symposium (QI-2008). pp. 133–140 (2008)
12. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Nlp (almost) from scratch. Journal of Machine Learning Research 12, 2493–2537 (2011)
13. Daiber, J., Jakob, M., Hokamp, C., Mendes, P.N.: Improving efficiency and accuracy in multilingual entity extraction. In: Procs of I-Semantics (2013)
14. Etzioni, O.: Machine reading at web scale. In: Proceedings of the 2008 International Conference on Web Search and Data Mining. pp. 2–2. WSDM '08 (2008)
15. Etzioni, O., Fader, A., Christensen, J., Soderland, S., Mausam, M.: Open information extraction: The second generation. In: Procs of IJCAI - Vol. 1. IJCAI'11 (2011)
16. Faruqui, M., Dodge, J., Jauhar, S.K., Dyer, C., Hovy, E., Smith, N.A.: Retrofitting word vectors to semantic lexicons. In: Proceedings of NAACL (2015)
17. Girju, R., Badulescu, A., Moldovan, D.: Learning semantic constraints for the automatic discovery of part-whole relations. In: Procs of the NAACL 2003 - Volume 1
18. Grefenstette, E., Sadrzadeh, M.: Experimental support for a categorical compositional distributional model of meaning. In: Procs of EMNLP 2011. pp. 1394–1404 (2011)
19. Harris, Z.: Distributional structure. Word 10(23), 146–162 (1954)
20. Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L.S., Weld, D.S.: Knowledge-based weak supervision for information extraction of overlapping relations. In: Proceedings of ACL 2011. pp. 541–550 (2011)
21. Hoffmann, R., Zhang, C., Weld, D.S.: Learning 5000 relational extractors. In: Proceedings of ACL 2010. pp. 286–295 (2010)
22. Kiela, D., Hill, F., Clark, S.: Specializing Word Embeddings for Similarity or Relatedness. Proceedings of EMNLP 2015 (September), 2044–2048 (2015)

23. Köhn, A.: What's in an Embedding? Analyzing Word Embeddings through Multilingual Evaluation. Proceedings of EMNLP 2015 (2014), 2067–2073 (2015)
24. Landauer, T.K., Dutnais, S.T.: A solution to platos problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. PSYCHOLOGICAL REVIEW 104(2), 211–240 (1997)
25. Le, Q., Mikolov, T.: Distributed Representations of Sentences and Documents. ICML pp. 1188–1196 (2014)
26. Liu, H., Singh, P.: Conceptnet — a practical commonsense reasoning tool-kit. BT Technology Journal 22(4), 211–226 (Oct 2004)
27. McAuley, J.J., Pandey, R., Leskovec, J.: Inferring networks of substitutable and complementary products. In: KDD (2015)
28. Mikolov, T., Corrado, G., Chen, K., Dean, J.: Efficient Estimation of Word Representations in Vector Space. Proceedings of ICLR 2013 (2013)
29. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
30. Miller, G.A.: Wordnet: A lexical database for english 38(11), 39–41 (Nov 1995)
31. Mitchell, J., Lapata, M.: Vector-based Models of Semantic Composition. Computational Linguistics (June), 236–244
32. Mooney, R.J.: Learning to connect language and perception. In: Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3. pp. 1598–1601. AAAI'08 (2008)
33. Navigli, R., Ponzetto, S.P.: Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial Intelligence 193(0), 217 – 250 (2012)
34. Radinsky, K., Agichtein, E., Gabrilovich, E., Markovitch, S.: A word at a time: Computing word relatedness using temporal semantic analysis. In: Proceedings of WWW2011. pp. 337–346. ACM (2011)
35. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proc. of the LREC Workshop on New Challenges for NLP Frameworks. pp. 45–50 (May 2010)
36. Reisinger, J., Mooney, R.J.: Multi-prototype vector-space models of word meaning. In: Procs of ACL 2010. pp. 109–117. Association for Computational Linguistics (2010)
37. Santos, C.D., Zadrozny, B.: Learning character-level representations for part-of-speech tagging. In: Proc. of the 31st ICML. pp. 1818–1826 (2014)
38. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: A core of semantic knowledge. In: Proceedings of WWW '07. pp. 697–706. ACM, New York, NY, USA (2007)
39. Sun, Y., Lin, L., Tang, D., Yang, N., Ji, Z., Wang, X.: Modeling mention, context and entity with neural networks for entity disambiguation. IJCAI International Joint Conference on Artificial Intelligence pp. 1333–1339 (2015)
40. Surdeanu, M., Tibshirani, J., Nallapati, R., Manning, C.D.: Multi-instance multi-label learning for relation extraction. In: Proceedings of EMNLP-CoNLL 2012. pp. 455–465 (2012)
41. 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)
42. Weston, J., Bordes, A., Yakhnenko, O., Usunier, N.: Connecting Language and Knowledge Bases with Embedding Models for Relation Extraction. EMNLP pp. 1366–1371
43. Xu, W., Hoffmann, R., Zhao, L., Grishman, R.: Filling knowledge base gaps for distant supervision of relation extraction. In: Proceedings of ACL 2013 - Volume 2: Short Papers. pp. 665–670 (2013)