Semantic Web 1 (2016) 1–5
IOS Press

Distributional and Neural Models for Extracting Manipulation-Relevant Relations from Text Corpora

Editor(s): Name Surname, University, Country
Solicited review(s): Name Surname, University, Country
Open review(s): Name Surname, University, Country

Soufian Jebbara [a,*], Valerio Basile [b,**], Elena Cabrio [b], and Philipp Cimiano [a]
[a] *CITEC, Bielefeld University, Inspiration 1, 33619, Bielefeld, Germany*
E-mail: {sjebbara,cimiano}@cit-ec.uni-bielefeld.de
[b] *Université Côte d'Azur, Inria, CNRS, I3S, Sophia Antipolis, France*
E-mail: valerio.basile@inria.fr; elena.cabrio@unice.fr

Abstract. In this paper we present a novel approach based on neural network techniques to extract common sense knowledge from text corpora. We apply this approach to extract of common sense knowledge about everyday objects that can be used by intelligent machines, e.g. robots, to support planning of tasks that involve object manipulation. The knowledge we extract is constituted by relations that relate some object (type) to some other entity such as its typical location or typical use. Our approach builds on the paradigm of distributional semantics and frames the task of extracting such relations as a ranking problem. Given a certain object, the goal is to extract a ranked list of locations or uses ranked by 'prototypicality'. This ranking is computed via a similarity score in the distributional space. We compare different techniques for constructing this semantic distributional space. On the one hand, we use the well known SkipGram model to embed words into a low-dimensional distributional space, using cosine similarity to rank the various candidates. We also consider an approach in the spirit of latent semantic indexing that relies on the NASARI approach to compute low-dimensional representations that are also ranked by cosine similarity. While both methods were already proposed in earlier work, as main contribution in this paper we present a neural network approach in which the ranking or scoring function is directly learned using a supervised approach. We compare all these approaches showing superiority of the neural network approach for some evaluation measures compared to the other two approaches described above for the construction of the distributional space.

Keywords: Relation Extration, Distributional Semantics, Supervised Learning, Commonsense Knowledge

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, to know where to put or store them, where and how to grasp them, and so on.

The field of cognitive robotics is concerned with how to endow machines with learning mechanisms that allow them to acquire such knowledge through experience [33]. However, acquiring manipulation-relevant knowledge requires many reproducible and similar experiences from which a system can learn how to manipulate a certain object. Some knowledge can

*Corresponding author. E-mail: sjebbara@cit-ec.uni-bielefeld.de
**Corresponding author. E-mail: valerio.basile@inria.fr

arguably even not be acquired by self-experience as relevant knowledge also comprises the mental properties that humans ascribe to certain objects. Such mental properties that are not intrinsic to the physical appearance of the object include for instance the intended use of an object. There are thus limits to what can be learned from self-guided experience with an object.

Further, letting every single robot or intelligent system acquire manipulation-relevant knowledge independently is not optimal from a resource-efficiency point of view. Instead, several projects worldwide have attempted to develop knowledge bases for robots through which knowledge, e.g. about how to manipulate certain objects, can be shared among many robots. Examples of such platforms are the RoboEarth project [55], RoboBrain [49] or KnowRob [53]

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. Work along these lines has, for instance, tried to derive plans on how to prepare a certain dish by machine reading descriptions of household tasks written for humans that are available on the web [54]. Other work has addressed the acquisition of scripts from the Web [45].

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.

We present and compare different approaches to extract manipulation-relevant knowledge from textual corpora. The knowledge extracted is of symbolic form and is not physically grounded [22]. Yet, this model can help robots or other intelligent systems to decide on how to act, support planning and select the appropriate models to manipulate a certain object.

Questions that such a knowledge base could answer include:

- Where should a certain object typically be stored?
- What is this object typically used for?
- Do I need to manipulate a certain object with care?

We treat the problem of extracting knowledge that can provide answers to such questions as a relation extraction problem. These relations include for instance the relation between an object and its prototypical location or an object and its typical intended use.

While there is a large body of work on relation extraction from text, we propose three novel contributions:

- In contrast to other approaches, we formulate the problem as a ranking problem. Given a certain object, our goal is to rank different targets as to how likely they are related to the given object in a particular fashion. This is a different task with respect to the one considered by standard relation extraction approaches that are mainly concerned with extracting every single mention of a relation from textual data.
- We experiment with different distributional spaces and show that both semantic vector spaces as considered within the NASARI approach as well as embedded word representations as produced by predictive language models such as skip grams provide already a reasonable performance on the task. A linear combination of both approaches has the potential to improve upon both approaches in isolation.
- We present a neural network approach that uses positive and negative examples of a relation to induce a scoring function that can be used to rank pairs. We show that this approach compares favourably to approaches in which a just the cosine similarity is used as measure of semantic relatedness to rank the pairs, but that it does not outperform the cosine similarity on all evaluation measures.
- Finally, we consider relations that are different from the relations that are typically considered in relation extraction, as our use case is to support intelligent systems to take decisions related to object manipulation. Standard relations considered in relation extraction are: *is-a, part-of, succession, reaction, production* [43,10] or *relation, parent/child, founders, directedBy, area_served, containedBy, architect, etc.* [47], or *albumBy, bornInYear, currencyOf, headquarteredIn, locatedIn, productOf, teamOf* [6] Instead, we consider manipulation-relevant relations such as the prototypical location and intended use of an object, henceforth denoted as `locatedAt` and `usedFor`, respectively.

The paper is structured as follows: In Section 2, we discuss related work from the fields of relation extraction, knowledge based population and knowledge

bases for robotics. In Section 3, we describe our approach to relation extraction in general and continue by introducing two models based on semantic relatedness as a ranking measure. These two models have been described in earlier work [5] and are described here again for the sake of completeness and due to the fact that we compare this previous work to a novel approach we introduce in Section 3.3. The model introduced in Section 3.3 is a supervised model that is trained to extract specific relations. Afterwards, in Section 4, we present our datasets that are used for training and testing the proposed models. We evaluate and compare all models in Section 5, showing that the supervised approach clearly outperforms the approaches based on semantic relatedness for some evaluation measures. Yet, the semantic relatedness approaches bear some value due to the fact that they do not have to be trained, but can be used off-the-shelf with any relation where the relation can be predicted from the types of the entities. In Section 6, we exploit insights gained from the evaluation to populate a knowledge base of manipulation-relevant data using the presented semi-automatic methods. Finally, in Section 7, we conclude our results and discuss directions for future work.

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 [21] and complete qualia structures describing an object [13]. 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 [19]). The DARPA Machine Reading Program [2] aims at endowing machines with capabilities for lifelong learning by automatically reading and understanding texts (e.g. [18]). 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. While the knowledge extract by our approach presented here is also not senso-motorically grounded, we hope that it can support planning of tasks involving object manipulation. 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 [41], 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. [10]). 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. [58,25,24,52]). Some researchers have modeled relation extraction as a matrix decomposition problem [47]. Other researchers have attempted to train relation extraction approach in a bootstrapping fashion, relying on knowledge available on the Web, e.g. [7].

Recently, scholars have tried to build models that can learn to extract generic relations from the data, rather than a set of pre-defined relations (see [31] and [8]). Such techniques are related to techniques to predict triples in knowledge graphs which rely on embeddings of entities (as vectors) and relations (as matrices) in the same distributional space (e.g. TransE [9] and TransH [56]). Similar ideas were tested in the past years in computational linguistics, where relations and modifiers are represented as tensors in the distributional space [4,17].

Ontologies and KB of common sense knowledge. DBpedia[1] 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 [51] is an ontology automatically extracted from WordNet and Wikipedia. YAGO extracts facts

[1] http://dbpedia.org

from the category system and the infoboxes of Wikipedia and combines these facts with taxonomic relations dervived from WordNet. Despite its high coverage, for our goals YAGO suffers from the same drawbacks of DBpedia, i.e. a lack of knowledge about common objects, that is, about their purpose, functionality, shape, prototypical location, etc.

ConceptNet[2] [32] 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[3] 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 the above resources are without doubt very useful resources, we are interested in developing an approach that can extract new knowledge from text corpora, complementing the knowledge contained in ontologies and knowledge bases such as the ones described above.

3. Extraction of Relations by a Ranking Approach based on Distributioanl Representations

This section presents our framework to extract relations between pairs of entities for the population of a knowledge base of manipulation-relevant data. We frame the task of relation extraction between entities as a ranking problem. Given a set of triples (h, r, t), where h is the head entity, r the relation and t the tail entity, we want to obtain a ranking of these triples. The triples in the upper positions of this ranking should be more likely correct than the triples in the lower parts.

Our general approach to produce these rankings is to design a scoring function $s(h, r, t)$ that assigns a score to each triple, depending on h, r, and t. The scoring function is designed in such a way that correct triples are assigned a higher score than incorrect triples. Sorting all triples by their respective scores produces the desired ranking. With a properly chosen function $s(h, r, t)$, it is possible to extract relations between entities to populate a knowlege base. This is achieved by scoring candidate triples and accept them or reject them based on their respective scores, e.g. if the score is above a certain threshold.

In this work, we present different scoring functions and evaluate them in the context of building a knowledge base of common sense triples. All of our proposed approaches rely on distributional representations of entities (and words). We investigate different vector representations and scoring functions, all with different strengths and weaknesses. In the follwing, we give a short introduction to distributional representations.

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. Traditionally, 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 similarity metric such as cosine similarity, defined as the cosine of the angle between two vectors:

$$similarity(\vec{x}, \vec{y})_{cos} = \frac{\vec{x} \cdot \vec{y}}{\| \vec{x} \| \| \vec{y} \|}$$

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" [23]. Several techniques can be applied to reduce the dimensionality of the cooccurrence matrix. Latent Semantic Analysis [29], 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.

Recently, neural network based models have received increasing attention in order to compute low-

[2] http://conceptnet5.media.mit.edu/

[3] http://www.opencyc.org/; as RDF representations: http://sw.opencyc.org/

dimensional representation of a certain input space. To compute representation of words, so called *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* [3].

Vector spaces created on word distributional representations have been successfully proven to encode word similarity and relatedness relations [44,46,14], and word embeddings have proven to be a useful feature in many natural language processing tasks [15,30, 48] in that they often encode semantically meaningful information of a word.

We argue that it is possible to extract interaction-relevant relations between entities, e.g. (`Object, locatedAt, Location`), using appropriate entity vectors and the cosine similarity since the domain and range of the considered relations are sufficiently narrow. In these cases, the semantic relatedness might be a good indicator for a relation.

3.1. Ranking by Cosine Similarity and Precomputed Word Embeddings

In the previous section, we motivated the use of distributional representations for the extraction of relations in order to populate a database of common sense knowledge. As outlined, we frame the relation extraction task as a ranking problem of triples (h, r, t) and score them based on corresponding vector representations $\boldsymbol{V}$.

In this section, we propose a neural network-based word embedding model to obtain distributional representations of entites. By using the relation-agnostic cosine similarity as our scoring function $s(h, r, t) = similarity_{cos}(\vec{v}_h, \vec{v}_t)$ we can interpret the vector similarity as a measure of semantic relatedness and thus as an indicator for a relation between the two entities.

Many word embedding methods encode useful semantic and syntactic properties [28,39,36] that we leverage for the extraction of locatedAt triples. In this work, we restrict our experiments to the skip-gram method [35]. 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 [39,36]. In particular, word representations of the skip-gram model often produce meaningful results using simple vector addition [36]. For this work, we trained the skip-gram model on a corpus of roughly 83 million Amazon reviews [34].

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 clean it by removing parts in parenthesis, if any, 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: $\vec{v}_{Public\_toilet} = \vec{v}_{public} + \vec{v}_{toilet}$. 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. By following this procedure for every entity in our dataset, we obtain a set of entity vectors $\boldsymbol{V}_{sg}$, derived from the original skip-gram word embeddings. With this derived set of entity vector representations, we can compute a score between pairs of entities based on the chosen scoring function, the cosine vector similarity. Using the example of `locatedAt`-pairs, this score is an indicator of how typical the location is for the object. 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). We refer to this model henceforth as `SkipGram/Cosine`.

Table 1

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

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

3.2. Ranking by Cosine Similarity and Semantically-Aware Entity Representations

Vector representations of words (Section 3.1) 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 [11], 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 [42], 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 cluster-based dimensionality reduction. The context of a concept is based on the set of Wikipedia pages where a mention of it is found. As shown by Camacho-Collados et al. [11], 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 (orthogonal, i.e. unrelated) to 1 (same direction, i.e. related).

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 Section 3, we focus on the extraction of interaction relevant relations by computing the cosine similarities of 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. We refer to this model as `NASARI/Cosine`.

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
	Furnace_room	.749
Paper_towel	Air_shower_(room)	.671
	Public_toilet	.634
	Bathroom	.632
	Mizuya	.597
	Kitchen	.589
Sump_pump	Furnace_room	.699
	Air_shower_(room)	.683
	Basement	.680
	Mechanical_room	.676

3.3. Ranking by a Trained Scoring Function

In the previous sections, we presented models of semantic relatedness for the extraction of relations. Since the employed cosine similarity function of these models is relation-agnostic, that is, it only measures whether there is a relation between two entities but not which relation in particular, the question that naturally arises is: *is it possible to predict* specific *given relations with a similar model?* In this section we try to answer this question by introducing a new model, based on supervised learning.

To extend the proposed approach to any kind of relation we modify the model presented in Section 3.1 by introducing a parameterized scoring function. This scoring function replaces the cosine similarity which was previously employed to score pairs of entities (e.g. Object-Location). By tuning the parameters of this new scoring function in a data-driven way, we are able to predict scores with respect to arbitrary relations.

We define the new scoring function $s(h,r,t)$ as a bilinear form:

$$s(h,r,t) = tanh(\vec{v}_h^\top \boldsymbol{M}_r \vec{v}_t + b_r) \quad (1)$$

where $\vec{v}_h, \vec{v}_t \in \mathbb{R}^d$ are the corresponding embedding vectors for the head and tail entities h and t, respectively, b_r is a bias term, and $\boldsymbol{M}_r \in \mathbb{R}^{d \times d}$ is the scoring matrix corresponding to the relation r. Our scoring function is closely related to the ones proposed by Jenatton et al. [26] as well as Yang et al. [59], however, we make use of the *tanh* activation function to map the

scores to the interval $(-1, 1)$. In part, this relates to the Neural Tensor Network in [50]. By initializing $\boldsymbol{M}_r$ as the identity matrix and b_r with 0, the inner term of the scoring function is effectively the dot product of $\vec{v}_h$ and $\vec{v}_t$ which is closely related to the originally employed cosine similarity.

We regularize our model by applying a Dropout layer [1] to the embedding vectors of the head and tail entity. We set the dropout fraction to 0.1, thus only dropping a small portion of the 100 dimensional input vectors.

The training of the scoring function is framed as a regression where we try to assign scores of 1 to all positive triples and scores of -1 to negative triples. We employ the mean squared error (MSE) as the training objective:

$$\begin{aligned}\mathcal{L} =& \frac{1}{|\mathcal{T}^+_{train}|} \sum_{(h,r,t)\in\mathcal{T}^+_{train}} (1 - s(h,r,t))^2 \\ &+ \frac{1}{|\mathcal{T}^-_{train}|} \sum_{(h,r,t)\in\mathcal{T}^-_{train}} (-1 - s(h,r,t))^2\end{aligned} \tag{2}$$

where $\mathcal{T}^+_{train}$ is the set of positive triples from our gold standard and $\mathcal{T}^-_{train}$ is a set of negative triples. During training, we keep the embedding vectors $\boldsymbol{V}$ fixed and only consider $\boldsymbol{M}_r$ and b_r as trainable parameters to measure the effect of the scoring function in isolation. Presumably, this allows for a better generalization to previously unseen entities.

Since our gold standard only provides us with the positive triples $\mathcal{T}^+_{train}$, we need to generate a set of corrupted negative triples $\mathcal{T}^-_{train}$. This is a procedure related to Noise Contrastive Estimation [40] and Negative Sampling [36] which is also used in the training of the SkipGram embeddings. We generate negative triples (h', r, t) and (h, r, t') for each positive triple $(h, r, t) \in \mathcal{T}^+$ by selecting negative head and tail entities h' and t' randomly from the set of all possible heads and tails, respectively. The exact number of negative triples that we generate per positive triple is a hyper-parameter of the model which we set to 10 triples[4] for all our experiments.

The supervised model improves upon the generic approaches in Sections 3.1 and 3.2 by its specificity to a certain relation. By training the model on one specific type of relation, e.g. the `locatedAt` relation from Section 4, we obtain a scoring function that is capable of measuring how well a given (unseen) pair of entities expresses this relation. In the following, we denote this model as `SkipGram/Supervised`.

[4] 5 triples (h', r, t) where we corrupt the head entity and 5 triples (h, r, t') where the tail entity is replaced.

4. Datasets

The following section introduces the datasets that we use for this work. We consider three datasests: i) a crowdsourced set of pairs standing in the `locatedAt` with human judgments, ii) a semi-automatically extracted set of pairs standing in the `locatedAt` relation, and iii) a semi-automatically extracted set of `usedFor` triples.

4.1. Crowdsourcing of Object-Location Rankings

In order to acquire valid pairs standing in the `locatedAt` relation we rely on a crowdsourcing approach. In particular, given a certain object, we used crowdsourcing to collect judgements about the likelihood to find this object at a set of predefined 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[5], 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 pairs standing in the `locatedAt` relation for the creation of the gold standard, we filter out odd or uncommon examples of objects or locations like `Ghodiyu` or `Fainting_room`. We order the objects by the number of incoming links to their respective Wikipedia page[6] in descending order and select the 100 top ranking objects for our gold standard. We proceed analogously for the locations, selecting 20 common locations and thus obtain 2,000 object-location pairs in total.

In order to collect the judgments, we set up a crowdsourcing experiment on the Crowdflower platform[7]. 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:

[5] Simple Knowledge Organization System: `https://www.w3.org/2004/02/skos/`

[6] We use the URI counts extracted from the parsing of Wikipedia with the DBpedia Spotlight tool for entity linking [16].

[7] `http://www.crowdflower.com/`

- **-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.
- **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 were 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 received at least 5 separate judgments, with some outliers collecting more than one hundred judgments each. The average agreement, i.e. the 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.

4.2. *Semi-Supervised Extraction of Object-Location Triples*

The SUN database [57] is a large-scale resource for computer vision and object recognition in images. It comprises 131.067 single images, each of them annotated with a label for the type of scene, and labels for each object identified in the scene. The images are 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 its original goal of providing high-quality data for training computer vision models, the SUN project generated a worth of semantic knowledge that is independent from the vision tasks. In particular, the labels are effectively semantic categories of entities such as objects and locations (scenes, using the lexical conventions of the SUN database).

Table 4
Most frequent pairs of object-scene in the SUN database.

Frequency	Object	Scene
1041	wall	b/bedroom
1011	bed	b/bedroom
949	floor	b/bedroom
663	desk_lamp	b/bedroom
650	night_table	b/bedroom
575	ceiling	b/bedroom
566	window	b/bedroom
473	pillow	b/bedroom
463	wall	b/bathroom
460	curtain	b/bedroom
406	painting	b/bedroom
396	floor	b/bathroom
393	cushion	b/bedroom
380	wall	k/kitchen
370	wall	d/dining_room
364	chair	d/dining_room
355	table	d/dining_room
351	floor	d/dining_room
349	cabinet	k/kitchen
344	sky	s/skyscraper

Objects are observed at particular scenes, and this relational information is retained in the database. In total, we extracted 31.407 object-scene pairs from SUN, together with the number of occurrences of each pair. The twenty most occurring pairs are shown in Table 4.

According to its documentation, the labels of the SUN database are lemmas from WordNet. However, they are not disambiguated and thus they could refer to any meaning of the lemma. Most importantly for our goals, the labels in their current state are not directly linked to any LOD resource. Faced with the problem of mapping the SUN database completely to a resource like DBpedia, we adopted a safe strategy for the sake of the gold standard creation. We took all the object and scene labels from the SUN pairs for which a resource in DBpedia with matching label exists. In order to limit the noise and obtain a dataset of "typical" location relations, we also removed from the pairs those that only occur once in the SUN database. This process resulted in 2,961 pairs of entities. We manually checked them and corrected 118 object labels and 44 location labels. In some cases the correct label was already present, so we eliminated the duplicates resulting in a new dataset of 2,935 object-location pairs.

4.3. Semi-Supervised Extraction of Object-Action Triples

While the methods we propose for relation extractions are by design independent of the particular relations they are applied to, we have focused most of our experimental effort towards one kind of relation between objects and locations, namely the typical location where given objects are found. In order to test the scalability of our approaches to other kind of relations, we created an alternative dataset revolving around a relation with the same domain as the location relation, i.e., objects, but a very different range, that is, actions. The relation under consideration will be refered to in the rest of the article as *usedFor*, for example the predicate *usedFor(soap, bath)* states that the soap is used for (or, during, in the process of) taking a bath.

We built a dataset of object-action pairs in a *usedFor* relation starting from ConceptNet 5 [32], a large semantic network of automatically collected commonsense facts (see also Section 2). From the entire ConceptNet, we extracted 46,522 links labeled *usedFor*. Although ConceptNet is partly linked to LOD resources, we found the coverage of such linking to be quite low, especially with respect to non-named entities such as objects. Therefore, we devised a strategy to link as many of the labels involved in *usedFor* relations to DBpedia, without risking to compromise the accuracy of such linking. The strategy is quite simple and it starts from the observation of the data: for the first argument of the relation, we search DBpedia for an entity whose label matches the ConceptNet labels. For the second argument, we search DBpedia for an entity label that matches the gerund form of the ConceptNet label, e.g., *Bath→Bathing*. We perform this step because we noticed how actions are usually referred to with nouns in ConceptNet, but with verbs in the gerund form in DBpedia. We used the morphology generation tool for English *morphg* [38] to generate the correct gerund forms also for irregular verbs. The application of this linking strategy resulted in a dataset of 1.674 pairs of DBpedia entities. Table 5 shows a few examples of pairs in the dataset.

Table 5
Examples of DBpedia entities in a *usedFor* relation, according to ConceptNet and our DBpedia linking strategy.

Object	Action
`Machine`	`Drying`
`Dictionary`	`Looking`
`Ban`	`Saving`
`Cake`	`Jumping`
`Moon`	`Lighting`
`Tourniquet`	`Saving`
`Dollar`	`Saving`
`Rainbow`	`Finding`
`Fast_food_restaurant`	`Meeting`
`Clipboard`	`Keeping`

5. Evaluation

This section presents the evaluation of the proposed framework for relation extraction (Sections 3.1, 3.2 and 3.3). We apply our models to the data described in Section 4, consisting of sets of (`Object`, `locatedAt`, `Location`) and (`Object`, `usedFor`, `Action`) triples. These experiments verify the feasability of our approach for the population of a knowledge base of manipulation relevant data.

We start our experiments by evaluating the produced rankings of Object-Location pairs with respect to a ground truth of human judgments. Secondly, we adapt the best performing method to automatically build a knowledge base and test its quality against the manually created gold standard dataset.

5.1. Ranking Evaluation

With the proposed methods, we are able to produce a ranking of e.g. locations for a given object that expresses how prototypical the location is for the given object. To test the validity of our methods we need to compare their output against a gold standard ranking. This gold standard ranking is extracted from the dataset described in Section 4.1 by assigning the average of the numeric values of the human judgments to each object-location pair. 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 first evaluation, we investigate how well the unsupervised baseline methods perform in creating object-location rankings. Secondly, we show how to improve these results by combining different approaches. Thirdly, we evaluate the supervised model in comparison to our baselines.

5.1.1. Unsupervised Object-Location Ranking Evaluation

Apart from the NASARI-based method (Section 3.2) and the SkipGram-based method (Section 3.1) we employ two simple baselines for comparison: For 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, the *link frequency*, is based on counting how often every object appears on the Wikipedia page of every location and vice versa. A ranking is produced based on these counts. An issue with this baseline is that the collected counts 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 two metrics: the *Normalized Discounted Cumulative Gain* (NDCG) and the *Precision at k* (Precision@k or P@k).

The NDCG is 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 accurately predict which locations are likely for a given object than to decide which are unlikely candidates.

While the NDCG measure gives a complete account of the quality of the produced rankings, 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, the *Precision@k*. The Precision@k measures the number of locations among the first k positions 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@k for $k = 1$ and $k = 3$ across all the objects.

Table 6 shows the average NDCG and Precision@k across all objects: methods NASARI/Cosine (Section 3.2) and SkipGram/Cosine (Section 3.1), plus the two baselines introduced above.

Table 6
Average Precision@k for $k = 1$ and $k = 3$ and average NDCG of the produced rankings against the gold standard rankings.

Method	NDCG	P@1	P@3
Location frequency baseline	.851	.000	.008
Link frequency baseline	.875	.280	.260
NASARI/Cosine	.903	**.390**	.380
SkipGram/Cosine	**.912**	.350	**.400**

Both our methods that are based on semantic relatedness outperform the simple baselines with respect to the gold standard rankings. 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 (Precision@1), as opposed to the better performance of SkipGram/Cosine in terms of Precision@3 and rank correlation.

We explored the results and found that for 19 objects out of 100, NASARI/Cosine correctly guesses the top ranking location where SkipGram/Cosine fails, while the opposite happens 15 out of 100 times. We also found that the NASARI-based method has a lower coverage than the SkipGram method, due to the coverage of the original resource (NASARI), where not every entity in DBpedia is assigned a vector[8]. The SkipGram-based method also suffers 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.

5.1.2. 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/Cosine, and at the same time experiment with hybrid methods to extract location relations, we devised two simple ways of combining the SkipGram/Cosine and NASARI/Cosine methods. The first method is based on a fallback strategy: given an ob-

[8]Objects like `Backpack` and `Comb`, and locations like `Loft` are all missing

Table 7

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

Method	NDCG	P@1	P@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	.423

ject, we consider the pair similarity of the object to the top ranking location according to NASARI/Cosine as a measure of confidence. If the top ranked location among the NASARI/Cosine ranking is exceeding a certain threshold, we consider the ranking returned by NASARI/Cosine as reliable. Otherwise, if the similarity is below the threshold, we deem the result unreliable and we adopt the ranking returned by SkipGram/Cosine instead. The second method produces object-location similarity scores by linear combination of the NASARI and SkipGram similarities. The similarity score for the generic pair h, t is thus given by:

$$\begin{aligned} sim_{\alpha}(h,t) = & \quad \alpha \cdot sim_{NASARI}(h,t) \\ + & \quad (1-\alpha) \cdot sim_{SkipGram}(h,t), \end{aligned} \tag{3}$$

where parameter α controls the weight of one method w.r.t. the other.

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/Cosine and SkipGram/-Cosine, for comparison. While the NDCG is basically not affected, both Precision@1 and Precision@3 show an increase in performance with Precision@3 showing the highest score of all investigated methods.

5.1.3. Supervised Object-Location Ranking

In the previous experiments, we investigated how well our (unsupervised) baseline methods perform when extracting the `locatedAt` relation. In the following, we compare the earlier results to the performance of a supervisedly trained scoring function. For this experiment we train the scoring function in Eq. (1)

Table 8

Average precision at k for $k = 1$ and $k = 3$ and average NDCG of the produced rankings against the gold standard rankings. *SkipGram/Supervised* denotes the supervised model based on SkipGram embeddings trained for the `locatedAt` relation.

Method	NDCG	P@1	P@3
Location frequency baseline	.851	.000	.008
Link frequency baseline	.875	.280	.260
NASARI/Cosine	.903	.390	.380
SkipGram/Cosine	.912	.350	.400
Linear combination (α=.4)	**.913**	.400	**.423**
SkipGram/Supervised	.908	**.454**	.387

to extract the `locatedAt` relation between objects and locations. The underlying embeddings $\boldsymbol{V}$ on which the scoring function computes its scores are fixed to the SkipGram embeddings $\boldsymbol{V}_{sg}$ (see Section 3.1). Here, the set of positive triples $\mathcal{T}^{+}_{train}$ is defined as the SUN triples that we extracted semi-automatically (see 4.2). From this, we generate the negative examples $\mathcal{T}^{-}_{train}$ following the procedure in Section 3.3.

As described in Section 3.3, we train the model by generating 10 negative triples per positive triple and minimizing the mean squared error (3). We initialize $\mathbf{M}_r$ with the identiy matrix, b_r with 0, and train the model parameter using stochastic gradient descent (SGD) using a learning rate of 0.001. SGD is performed in mini batches of size 100 with 300 epochs of training. The training procedure is realized with *Keras* [12].

As before, we test the model on the curated set of objects and locations described in Section 6 and produce a ranking of locations for each object. Table 8 shows the performance of the extended model (SkipGram/Supervised) in comparison to the previous approaches.

Overall, we can observe mixed results. All of our proposed models improve upon the baseline methods with respect to all evaluation metrics. Compared to the SkipGram/Cosine model, the SkipGram/Supervised model decreases slightly in performance with respect to the NDCG and more so for the Precision@3 score. Most striking, however, is the increase in Precision@1 of SkipGram/Supervised, showing a relative improvement of 30% to the SkipGram/Cosine model and constituting the highest overall Precision@1 score by a large margin. However, the linear combination (α=.4) still scores higher with respect to Precision@3 and NDCG.

While the presented results do not point to a clear preference for one particular model, Section 6 will in-

vestigate the above methods more closely in the context of the generation of a knowledge base.

5.2. Retrieval Evaluation

In the previous section, we tested how the proposed methods perform 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 manipulation-relevant information about objects in an automatic fashion.

All the methods introduced in this work are based on some scoring function of triples 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 two gold standard datasets introduced in Sections 4.1 and 4.3, covering two different relations, i) the `locatedAt` relation between objects and locations that we saw in earlier experiments and ii) the `usedFor` relation between objects and actions. We introduce the `usedFor` relation in order to verify the generalizability of our supervised scoring function.

As the gold standard data for the `locatedAt` relation, we use the version with data aggregated by Crowdflower: the contributors' 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.

In order to evaluate the actual retrieval performance of our ranking based methods with respect to the population of a knowledge base, we score the test sets for each dataset with our presented methods and order the produced lists by the generated scores.

In general, we order the entity pairs produced by our methods by similarity score, and select the first k from the list, with k being a parameter. We evaluate the retrieved lists in terms of Precision, Recall and F-score against the gold standard sets with varying values of k. 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. The F-score is the harmonic mean of precision and recall.

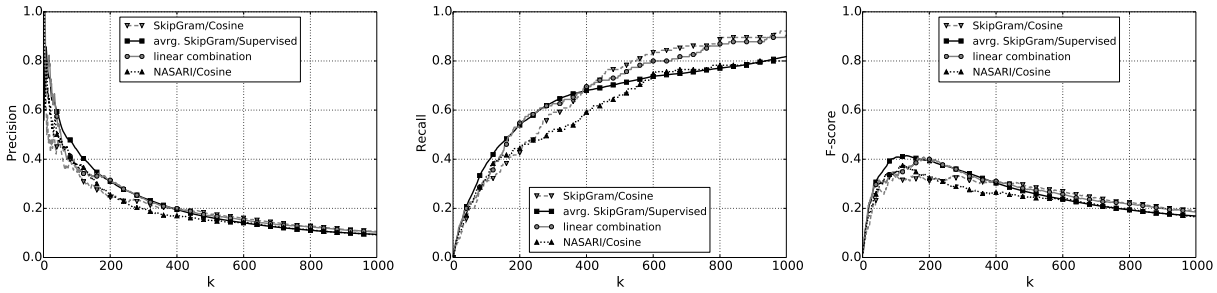
For the `locatedAt` relation, we also add to the comparison the results of the hybrid, linear combination method from Section 5.1.2, with the best performing parameters in terms of Precision@1, namely the linear combination with $\alpha = 0.4$.

Figures 1 and 2 show the evaluation of the four methods evaluated against the two aggregated gold standard datasets for the `locatedAt` relation described above. Figures 1c and 2c, in particular, show F-score plots for a direct comparison of the performance. The SkipGram/Supervised model achieves the highest F-score in the "usual" setting, peaking at $k = 132$ with an F-score of 0.415. The SkipGram/Cosine model and the linear combination outperform both the NASARI/Cosine and the SkipGram/Supervised in terms of recall, especially for higher k. This also holds for the "usual+plausible" locations. Here, the SkipGram/Supervised model stands out by achieving high precision values for small values of k. Overall, SkipGram/Supervised performs better for small k (50 – 400) whereas SkipGram/Cosine and the linear combination obtain better results with increasing k. This seems to be in line with the results from previous experiments in Table 8 that show a high Precision@1 for the SkipGram/Supervised model but higher scores for SkipGram/Cosine and the linear combination in terms of Precision@3.

5.3. Evaluation of Object-Action pairs extraction

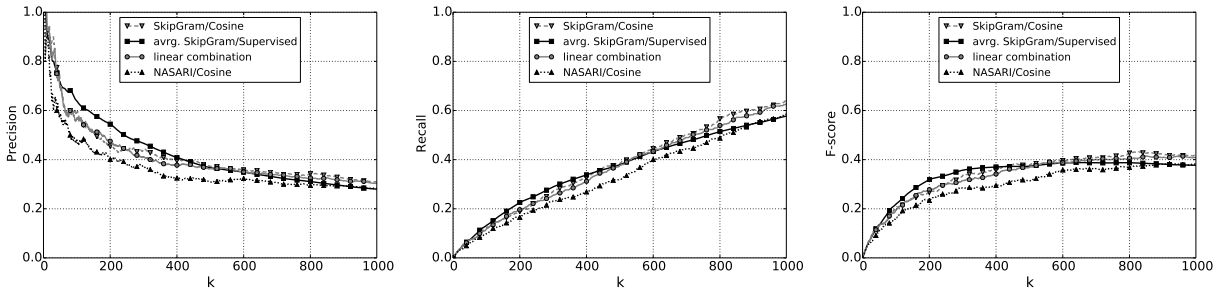
One of the reasons to introduce a novel technique for relation extraction based on supervised statistical method, as stated previously, is to be able to scale the extraction across different types of relations. To test the validity of this statement, we apply the same evaluation procedure introduced in the previous part of this section to the `usedFor` relation.

The gold standard for the `usedFor` relation is obtained from the ConceptNet dataset (see Section 4.3)



(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).

by randomly dividing the complete set of positive `(Object, usedFor, Action)` triples in a training portion (527 triples) and a test portion (58 triples). We combine each object entity in the test portion with each action entity to generate a complete test set, comprised of positive and negative triples[9]. To account for variations in the performance due to this random partitioning, we repeat each experiment 100 times and report the averaged results. The average size of the test set is ≈ 2059.

Figure 3 displays precision, recall and F-score for retrieving the top k results. Again, the results are averaged scores over 100 experiments to account for variations in performance due to the random partitioning in training and test triples and the generation of negative samples. The standard deviation for precision, recall and F-score for all k is visualized along the mean scores.

The supervised model achieves on average a maximum F-score of about 0.465 when extracting 70 triples. This is comparable to the achieved F-Scores when training the scoring function for the `locatedAt` relation. To give an insight into the produced false positives, Table 9 shows the top 30 extracted triples for

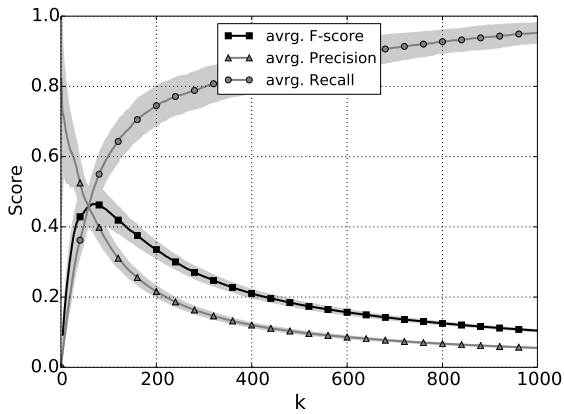


Fig. 3. Evaluation of knowledge base generation for the `usedFor` relation between objects and actions. Precision, Recall and F-score a given with respect to extracting the to k scored triples.

[9]We filter out all generated triples that are falsely labeled as negative in this process.

Table 9
A list of the top 30 extracted triples for the `usedFor` relation. The gray highlighted rows mark the entity pairs that are part of the gold standard dataset (Section 4.3).

Score	Object	Action
1.00000	Snack	Snacking
0.99896	Snack	Eating
0.99831	Curry	Seasoning
0.99773	Drink	Drinking
0.98675	Garlic	Seasoning
0.98165	Oatmeal	Snacking
0.98120	Food	Eating
0.96440	Pistol	Shooting
0.95218	Drink	Snacking
0.94988	Bagel	Snacking
0.94926	Wheat	Snacking
0.93778	Laser	Printing
0.92760	Food	Snacking
0.91946	Typewriter	Typing
0.91932	Oatmeal	Eating
0.91310	Wok	Cooking
0.89493	Camera	Shooting
0.85415	Coconut	Seasoning
0.85091	Stove	Frying
0.85039	Oatmeal	Seasoning
0.84038	Bagel	Eating
0.83405	Cash	Gambling
0.81985	Oatmeal	Baking
0.80975	Lantern	Lighting
0.80129	Calculator	Typing
0.78279	Laser	Shooting
0.77411	Camera	Recording
0.75712	Book	Writing
0.72924	Stove	Cooking
0.72280	Coconut	Snacking

the `usedFor` relation of one trained instance of the supervised model.

6. Building a Knowledge Base of Object Locations

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 4.1), 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]

The knowledge base created with this method is the result of one among many possible configurations of a number of methods and parameters. In particular, the creator of a knowledge base involving the extraction of relations is given the choice to prefer precision over recall, or vice-versa. This is done, in our method, by adjusting the threshold on the similarity scores. Employing different algorithms for the computation of the actual similarities (word embeddings vs. entity-word vectors, supervised vs. unsupervised models) are also expected to result in different knowledge bases. A qualitative assessment of such impact is left for future work.

7. Conclusion and Future Work

We have presented a framework for extracting manipulation-relevant knowledge about objects in the form of (binary) relations. The framework relies on a ranking measure that, given an object, ranks all entities that potentially stand in the relation in question to the given object. We rely on a representational approach that exploits distributional spaces to embed entities into low-dimensional spaces in which the ranking measure can be evaluated. We have presented results on two relations: the relation between an object and

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

its prototypical location (`locatedAt`) as well as the relation between an object and one of its intended uses (`usedFor`). We have shown that both an approach relying on standard embeddings computed by a Skip-Gram model as well as embeddings computed by a semantic dimensionality reduction approach (NASARI) perform very well compared to two rather naive baselines. Both approaches were presented already in previous work. As main contribution of this paper, we have presented a supervised approach based on a neural network that, instead of using the cosine measure as measure of semantic relatedness, uses positive and negative examples to train a scoring function in a supervised fashion. In contrast to the other two unsupervised approaches, the latter model learns a model that is specific for a particular relation while the other two approaches implement a general notion of semantic relatedness in distributional space. We have shown that the improvements of the supervised model are not always clear compared to the two unsupervised approaches. This might be attributable to the fact that the types of both relations (usedFor and locatedAt) are specific enough to predict the relation in question. Whether the unsupervised approach would generalize to relations with a less specific type signature is a question for future work.

As future work, we would like to employ *retrofitting* [20] to enrich our pretrained word embeddings with concept knowledge from a semantic network such as ConceptNet or WordNet [37] 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 [27] even better results could be achieved.

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.

Acknowledgments The authors wish to thank the anonymour reviewers of EKAW, who provided useful feedback to make this extended version of the paper. The work in this paper is partially funded by the ALOOF project (CHIST-ERA program) and by the Cluster of Excellence Cognitive Interaction Technology 'CITEC' (EXC 277), Bielefeld University.

References

[1] Dropout : A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research (JMLR)*, 15:1929–1958, 2014.

[2] Ken Barker, Bhalchandra Agashe, Shaw-Yi Chaw, James Fan, Noah Friedland, Michael Glass, Jerry Hobbs, Eduard Hovy, David Israel, Doo Soon Kim, Rutu Mulkar-Mehta, Sourabh Patwardhan, Bruce Porter, Dan Tecuci, and Peter Yeh. Learning by reading: A prototype system, performance baseline and lessons learned. In *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 1*, AAAI'07, pages 280–286, 2007.

[3] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of ACL 2014 (Volume 1: Long Papers)*, June 2014.

[4] Marco Baroni and Roberto Zamparelli. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1183–1193, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[5] Valerio Basile, Soufian Jebbara, Elena Cabrio, and Philipp Cimiano. Populating a Knowledge Base with Object-Location Relations Using Distributional Semantics. In *20th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2016)*, pages 34 – 50, Bologna, Italy, November 2016.

[6] Sebastian Blohm and Philipp Cimiano. Using the web to reduce data sparseness in pattern-based information extraction. In *Knowledge Discovery in Databases: PKDD 2007, 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, Warsaw, Poland, September 17-21, 2007, Proceedings*, pages 18–29, 2007.

[7] Sebastian Blohm, Philipp Cimiano, and Egon Stemle. Harvesting relations from the web - quantifiying the impact of filtering functions. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, pages 1316–1321, 2007.

[8] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1247–1250, New York, NY, USA, 2008. ACM.

[9] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2787–2795. Curran Associates, Inc., 2013.

[10] Razvan C. Bunescu and Raymond J. Mooney. Subsequence kernels for relation extraction. In *Advances in Neural Information Processing Systems 18 [Neural Information Processing Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada]*, pages 171–178, 2005.

[11] José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. In *HLT-NAACL (2015)*.

[12] François Chollet. Keras. `https://github.com/fchollet/keras`, 2015.

[13] Philipp Cimiano and Johanna Wenderoth. Automatically learning qualia structures from the web. In *Proceedings of the ACL-SIGLEX Workshop on Deep Lexical Acquisition*, DeepLA '05, pages 28–37, 2005.

[14] Alina Maria Ciobanu and Anca Dinu. Alternative measures of word relatedness in distributional semantics. In *Joint Symposium on Semantic Processing.*, page 80, 2013.

[15] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. NLP (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

[16] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of I-Semantics*, 2013.

[17] Tim Van de Cruys, Thierry Poibeau, and Anna Korhonen. A tensor-based factorization model of semantic compositionality. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 1142–1151, 2013.

[18] Oren Etzioni. Machine reading at web scale. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, pages 2–2, 2008.

[19] Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam Mausam. Open information extraction: The second generation. In *Proceedings of IJCAI - Vol. 1*, IJCAI'11, 2011.

[20] Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. Retrofitting word vectors to semantic lexicons. In *Proceedings of NAACL*, 2015.

[21] Roxana Girju, Adriana Badulescu, and Dan Moldovan. Learning semantic constraints for the automatic discovery of part-whole relations. In *Proceedings of the NAACL 2003 - Volume 1*.

[22] Stevan Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42:335–346, 1990.

[23] Zellig Harris. Distributional structure. *Word*, 10(23):146–162, 1954.

[24] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke S. Zettlemoyer, and Daniel S. Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of ACL 2011*, pages 541–550, 2011.

[25] Raphael Hoffmann, Congle Zhang, and Daniel S. Weld. Learning 5000 relational extractors. In *Proceedings of ACL 2010*, pages 286–295, 2010.

[26] Rodolphe Jenatton, Nicolas L. Roux, Antoine Bordes, and Guillaume R Obozinski. A latent factor model for highly multi-relational data. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 3167–3175. Curran Associates, Inc., 2012.

[27] Douwe Kiela, Felix Hill, and Stephen Clark. Specializing Word Embeddings for Similarity or Relatedness. Number September, pages 2044–2048, 2015.

[28] Arne Köhn. What's in an Embedding? Analyzing Word Embeddings through Multilingual Evaluation. Number 2014, pages 2067–2073, 2015.

[29] Thomas K Landauer and Susan T. Dutnais. A solution to platoâĂŹs problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *PSYCHOLOGICAL REVIEW*, 104(2):211–240, 1997.

[30] Qv Le and Tomas Mikolov. Distributed Representations of Sentences and Documents. *ICML*, pages 1188–1196 (2014).

[31] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of AAAI 15*, 2015.

[32] H. Liu and P. Singh. Conceptnet –a practical commonsense reasoning tool-kit. *BT Technology Journal*, 22(4):211–226, October 2004.

[33] Max Lungarella, Giorgio Metta, Rolf Pfeifer, and Giulio Sandini. Developmental robotics: a survey. *Connect. Sci.*, 15(4):151–190, 2003.

[34] J. J. McAuley, R. Pandey, and J. Leskovec. Inferring networks of substitutable and complementary products. In *KDD*, 2015.

[35] Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *Proceedings of ICLR 2013*, 2013.

[36] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[37] George A. Miller. Wordnet: A lexical database for english. 38(11):39–41, November 1995.

[38] Guido Minnen, John Carroll, and Darren Pearce. Applied morphological processing of english. *Nat. Lang. Eng.*, 7(3):207–223, September 2001.

[39] Jeff Mitchell and Mirella Lapata. Vector-based Models of Semantic Composition. *Computational Linguistics*, (June):236–244.

[40] Andriy Mnih and Yee Whye Teh. A fast and simple algorithm for training neural probabilistic language models. In *In Proceedings of the International Conference on Machine Learning*, 2012.

[41] Raymond J. Mooney. Learning to connect language and perception. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, AAAI'08, pages 1598–1601, 2008.

[42] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193(0):217 – 250, 2012.

[43] Patrick Pantel and Marco Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In Nicoletta Calzolari, Claire Cardie, and Pierre Isabelle, editors, *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006*. The Association for Computer Linguistics, 2006.

[44] Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. A word at a time: Computing word relatedness using temporal semantic analysis. In *Proceedings of WWW2011*, pages 337–346. ACM, 2011.

[45] Michaela Regneri, Alexander Koller, and Manfred Pinkal. Learning script knowledge from web experiments. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, Uppsala, 2010.

[46] Joseph Reisinger and Raymond J Mooney. Multi-prototype vector-space models of word meaning. In *Proceedings of ACL 2010*, pages 109–117. Association for Computational Linguistics, 2010.

[47] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. Relation extraction with matrix factorization and universal schemas. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, editors, *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 74–84. The Association for Computational Linguistics, 2013.

[48] Cicero D Santos and Bianca Zadrozny. Learning character-level representations for part-of-speech tagging. In *Proc. of the 31st ICML*, pages 1818–1826, 2014.

[49] Ashutosh Saxena, Ashesh Jain, Ozan Sener, Aditya Jami, Dipendra Kumar Misra, and Hema Swetha Koppula. Robobrain: Large-scale knowledge engine for robots. *CoRR*, abs/1412.0691, 2014.

[50] Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. Reasoning With Neural Tensor Networks for Knowledge Base Completion. 2013.

[51] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A core of semantic knowledge. In *Proceedings of WWW '07*, pages 697–706, New York, NY, USA, 2007. ACM.

[52] Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. Multi-instance multi-label learning for relation extraction. In Jun'ichi Tsujii, James Henderson, and Marius Pasca, editors, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 455–465. ACL, 2012.

[53] Moritz Tenorth and Michael Beetz. Knowrob: A knowledge processing infrastructure for cognition-enabled robots. *I. J. Robotics Res.*, 32(5):566–590, 2013.

[54] Moritz Tenorth, Daniel Nyga, and Michael Beetz. Understanding and executing instructions for everyday manipulation tasks from the world wide web. In *IEEE International Conference on Robotics and Automation, ICRA 2010, Anchorage, Alaska, USA, 3-7 May 2010*, pages 1486–1491, 2010.

[55] Markus Waibel, Michael Beetz, Raffaello D'Andrea, Rob Janssen, Moritz Tenorth, Javier Civera, Jos Elfring, Dorian Gálvez-López, Kai Häussermann, J.M.M. Montiel, Alexander Perzylo, Björn Schießle, Oliver Zweigle, and René van de Molengraft. RoboEarth - A World Wide Web for Robots. *Robotics & Automation Magazine*, 18(2):69–82, 2011.

[56] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In Carla E. Brodley and Peter Stone, editors, *AAAI*, pages 1112–1119. AAAI Press, 2014.

[57] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, pages 3485–3492. IEEE Computer Society, 2010.

[58] Wei Xu, Raphael Hoffmann, Le Zhao, and Ralph Grishman. Filling knowledge base gaps for distant supervision of relation extraction. In *Proceedings of ACL 2013 - Volume 2: Short Papers*, pages 665–670, 2013.

[59] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. *Iclr*, page 12, 2015.