Assessing the Lexico-Semantic Relational Knowledge Captured by Word and Concept Embeddings

Ronald Denaux, Jose Manuel Gomez-Perez

Expert System Cogito Labs {rdenaux,jmgomez}@expertsystem.com

Abstract

Deep learning currently dominates the benchmarks for various NLP tasks and, at the basis of such systems, words are frequently represented as embeddings-vectors in a low dimensional space-learned from large text corpora and various algorithms have been proposed to learn both word and concept embeddings. One of the claimed benefits of such embeddings is that they capture knowledge about semantic relations. Such embeddings are most often evaluated through tasks such as predicting humanrated similarity and analogy which only test a few, often ill-defined, relations. In this paper, we propose a method for (i) reliably generating word and concept pair datasets for a wide number of relations by using a knowledge graph and (ii) evaluating to what extent pre-trained embeddings capture those relations. We evaluate the approach against a proprietary and a public knowledge graph and analyze the results, showing which lexico-semantic relational knowledge is captured by current embedding learning approaches.

1 Introduction and Related Work

Most of the recent interest in the area of word embeddings was triggered by the Word2Vec algorithm proposed in [Mikolov et al., 2013], which provided an efficient way to learn word embeddings by predicting words based on their context and using negative sampling. Word embeddings have become the usual input to natural language processing (NLP) tasks, but also tasks for which previously knowledge graphs were being used. Applications range from text classification [Kim, 2014] to machine translation [Kalchbrenner and Blunsom, 2013; Sutskever et al., ; Cho et al., 2014; Bahdanau et al., 2014], question answering [Khot et al., ; Seo et al., 2016; Parikh,] and knowledge graph construction and completion[Fu et al., 2014]. However, despite recent efforts [Li et al., ; Garcia and Gomez-Perez, 2018; Zeiler and Fergus, 2014], the nature and extent of the knowledge captured by such embeddings and how they contribute to accomplish the goal in question is still hard to interpret.

Embeddings have shown the ability to learn relations between words. However, most benchmarks [Schnabel *et al.*, 2015] focus on a specific family of relations involving relations similarity and analogy. It is unclear whether such relations simply happen to be well aligned with the statistic analysis involved in the computation of the embeddings, or whether the embeddings are capable of capturing a wider range of relational knowledge. Indeed, standard benchmarks show evidence that word embeddings may capture more types of relations. It has been shown that algorithms like FastText [Bojanowski *et al.*, 2017], GloVe [Pennington *et al.*, 2014] and Swivel [Shazeer *et al.*, 2016] learn embeddings that capture lexical and semantic information. However, the current lack of a standard evaluation practice make it hard to study which specific relations embeddings can effectively capture, nor how to best quantify the signal of such relations.

At the same time, embeddings as a knowledge representation mechanism is being explored by the traditionally symbolic community that produced semantic networks and knowledge graphs. Algorithms based on knowledge graph (KG) embeddings, like RDF2Vec[Ristoski and Paulheim, 2016], ProjE[Shi and Weninger, 2017], TransE [Bordes et al., 2013] and HolE[Nickel et al., 2016b], learn embeddings representing the concepts, words and relations contained in a KG and provide a vector representation of the knowledge that is explicitly described in it. However, such KG embeddings may only encode knowledge that is already represented in the KG. One application of embeddings for this community is generic entity-based KG completion and refinement [Riedel et al., 2013; Melo and Paulheim, 2017; Nickel et al., 2016a; Paulheim, 2017; Lin et al., 2016], which tries to use large text corpora to complete a partial knowledge graph. However, such efforts have focused on encyclopedic (e.g. DBpedia) or domain-specific (family, commerce, finance, law) relations between entities rather than lexical relations. On the opposite direction, KGs are also used to refine vector space representations as in [Faruqui et al., 2015]. Finally, efforts trying to learn lexical semantic relations [Shwartz and Dagan, 2016; Gábor et al., 2017; Turney and Mohammad, 2015; Roller et al., 2014; Fu et al., 2014] have started looking at whether word embeddings (and similar distributional approaches) can be used to predict certain types of lexical semantic relations. Many of these approaches suffer from the difficulty to generate a training dataset that serves this purpose [Levy et al., 2015] and most of them focus on hypernymy relations or lexical inference, still a limited fragment of the whole spectrum of possible lexical semantic relations. Also, WordNet tends to be the only source of evidence used in such work, which may hinder reaching a general understanding of the matter.

In previous work[Denaux and Gómez-Pérez, 2017; Denaux and Gomez-Perez, 2019]1 on joint word-concept embeddings, these outperformed word-only and knowledge graph approaches over a selection of 14 benchmarks on semantic similarity and relatedness as well as word-concept and hypernym relation prediction tasks. In doing so, we suggested it may be possible to extract knowledge about lexical semantic relations from (joint word-concept) embeddings. In this paper, we take into account lessons from the various communities discussed above to propose a method for evaluating to what extent pre-trained embeddings contain knowledge about a wide ranging of relations encoded in a KG. Our contributions are: (i) describing a (largely automated) method for (i.a) word/concept relational dataset generation from a KG (i.b) training of machine learning models and (i.c) evaluation of the generated datasets and trained models; (ii) application of this method to analyse to what extent (ii.a) various corpusand KG-based pre-trained embeddings capture (ii.b) various types of lexico-semantic relational knowledge and (ii.b) what the effect is various factors such as the size of the corpus and the type of dataset (word or concept).

In Section 2 we describe a generic method for measuring the predictive power of embeddings for specific relations. In section 3 we describe how we apply this methodology to study lexico-semantic relations and in section 4 we analyse the gathered data.

2 Measuring Relation Predictive Power of Embeddings with a KG

In this section we propose a method for studying the predictive power of word and concept embeddings, depicted in Figure 1. The goal is to study how much knowledge about relations is encoded in embedding spaces and can be exploited by machine learning models. Note that our goal is *not* to generate the best models for predicting relations (which could be achieved by combining different sources of evidence). The method consists of three main phases: dataset generation, model training and prediction results analysis.

Preliminaries We define a **Knowledge Graph** as a tuple $\langle N, R, E \rangle$, where $N = C \cup I \cup W$ is a set of node identifiers, typically referring to concepts $c \in C$, instances $i \in I$ or words $w \in W$ (human readable names for concepts and instances); R is a set of relation types and E is a set of triples of the form $(r \ n_i \ n_j)$ where $r \in R$ and $n_i, n_j \in N$. We define an **embedding space** S as a tuple $\langle V, F_d \rangle$, where V is a vocabulary, and $F_d : V \longrightarrow \mathbb{R}^d$ is a function that maps elements in the vocabulary to its embedding:

The main *inputs* for our method are a KG $k = \langle N_k, R_k, E_k \rangle$ and one or more embedding spaces $s_i \in S'_k \subset S_k \subset S$. S_k is the set of all embedding spaces that have a vocabulary that overlaps with N_k . S'_k is a subset of embedding spaces to study. The main *output* of our method is a

classification of tuples $(r \ s)$, where $r \in R_k$ and $s \in S'_k$, into *predictable* and *non-predictable*. Furthermore, for each such tuple, we also derive absolute and relative relation prediction metrics, providing a numerical assessment of how well the embedding space s captures or encodes knowledge about relationship r.

Dataset generation In the first phase, we aim to generate datasets $\delta \in D$, where each dataset δ is a finite set of triples of the form $\delta_r = \{ \langle n_i \ n_j \ l \rangle \}$, where l is a classification label such that l = 1 if $(r \ n_i \ n_j) \in E_k$ and l = 0 otherwise. We generate datasets between *words* δ^w with tuples $\langle w_i \ w_j \ l \rangle$, word-concepts δ^{wc} with tuples $\langle w_i \ c_j \ l \rangle$ and concepts δ^c with tuples $\langle c_i \ c_j \ l \rangle$.

As a first step, we select a **seed vocabulary** $V_{\text{seed}} = \bigcap_{i=1}^{|S'_k|} V_{s_i}$, as the intersection of all the studied vocabularies. Next, V_{seed} and k are used to **extract a set of positive pairs** for each $r \in R_k$; these correspond to partial datasets $\delta_r^+ = \{ \langle n_i \ n_j \ 1 \rangle \mid (r \ n_i \ n_j) \in E_k \land n_i, n_j \in V_{\text{seed}} \}$. KGs frequently define relations at a concept level, therefore, this initial dataset will typically only contain concepts. To also generate word-concept datasets for such relations, we also extract partial datasets $\{ \langle w_i \ c_j \ 1 \rangle \mid (r \ c_i \ c_j) \land (r_w \ w_i \ c_i) \}$, where r_w is the word-to-concept relation in k. Similarly, we can extract a word dataset for r using $\{ \langle w_i \ w_j \ 1 \rangle \mid (r \ c_i \ c_j) \land (r_w \ w_i \ c_i) \}$.

Besides generating positive datasets for relations $r \in R_k$, we also generate *random datasets*. These datasets serve as baselines and will be used later on to prune biased datasets. Since the generated datasets δ_r^+ vary in size depending on the number of relation tuples in E_k , we generate sets of varying sizes ² of "positive" random pairs $\{\langle n_i n_j 1 \rangle\}$, where n_i and n_j are randomly sampled from N_k . We refer to these datasets as $\delta_{\text{rand},x}^+$ where x is the number of positive pairs generated.

We need both positive and negative examples in the datasets to train a model. For ease of training, we aim to generate balanced datasets with the same number of positive and negative examples. Randomly generating negative pairs based on the seed vocabulary or selecting positive pairs of a different relation type is not optimal because models can learn to identify words/concepts associated to the relation rather than the relation itself [Levy et al., 2015]. Instead, we apply negative switching, whereby positive pairs are switched based on the subject and object vocabularies for the relation, i.e. from positive pairs $\langle n_i \ n_j \ 1 \rangle$ and $\langle n_k \ n_l \ 1 \rangle$; thus subject vocabulary $\{n_i, n_k\}$ and object vocabulary $\{n_i, n_l\}$, it is possible to generate negative pairs $\langle n_k n_i 0 \rangle$ and $\langle n_i n_l 0 \rangle$. Albeit practical, this approach does not take into account transitive relations and assumes a closed world. Also, for relations with unbalanced subject-object vocabularies, it may be impossible to generate sufficient negative examples; in such cases we fall back to selecting pairs from other relation types or generating random pairs.

Model training We next use the generated datasets to train binary classification machine learning models for each studied embedding space $s \in S'_k$, we refer to the resulting trained models as $m^{\delta,s,t}$, where δ is the generated dataset and t is the

¹Vecsigrafo: Corpus-based Word-Concept Embeddings

 $^{^{2}}$ In this work we use sizes 200, 500, 1K, 5K, 10K and 50K.



Figure 1: Generic approach for evaluating the predictive power of word and concept embeddings using a knowledge graph as a silver standard.

model type. We do not specify a type of machine-learning model to be used; however, since the input and output are very simple, we expect fully-connected neural networks to be suitable and use these in our experiments.

Each sample $\langle n_i n_j l \rangle$ in the dataset is converted into an *in*put vector for the model by combining the embeddings for the subject and object arguments; i.e. $F_s(n_i) \odot F_s(n_j)$. In this work we use vector concatenation for \odot , but other operations are also possible. However, to prevent models from simply learning which embeddings are associated with a relation type, during training we apply input perturbation: for each batch, we generate a random vector $v \in \mathbb{R}^d$ and add it to both the subject and object embedding. Thus the final input to the model is $(F_s(n_i) \oplus v) \odot (F_x n_i \oplus v)$. This ensures that the difference between the embeddings is the same as for the original pair, but no individual embedding is seen twice. Also, since different embedding spaces have different scales, we adapt the amount of perturbation to each embedding space by scaling the random perturbation to be within the standard deviation of the embedding space.

Furthermore, to verify that the generated dataset does not encode information about the relation, we also train models using a baseline random embedding space $s_{rand} \in S_k$ for the seed vocabulary. Since the embeddings in s_{rand} are random, they cannot encode any information about $n \in V_{seed}$ and their relations $r \in R_k$.

Each generated dataset is split into training, (validation) and testing subsets; the latter is used to evaluate the trained model by calculating metrics: precision, recall, accuracy and f1. Since the performance of a model is affected by random initialization of its parameters, we propose to perform multiple runs, resulting in $m_i^{\delta,s,t}$ for $1 \le i \le \#_{\text{runs}}$. This allows us to calculate the average and standard deviation for each of the collected metrics. Below, we use $\mu_{\delta,s}^v$ to refer to the average for metric v for models trained on dataset δ and embedding space s. If s is omitted, the average is taken over all models trained on δ . Likewise for the standard deviation σ^v .

Prediction results analysis Once we have trained all our models, we can do some statistic analysis on the collected metrics. Since the dataset generation and model training are

mostly automated, our main goal here is to identify any nonpredictable relations. Using our baselines, we can discard results based on two reasons: (i) the generated dataset is biased and (ii) the learned model's results are not significant.

First, we define **baseline ranges for prediction met**rics. For this, we use the metrics gathered for the generated random datasets $\delta_{\text{rand},x}$ and define the range thresholds as $\tau_{\text{biased}}^{\upsilon_{\min}} = \mu_{\delta_{\text{rand},x}}^{\upsilon} - 2\sigma_{\delta_{\text{rand},x}}^{\upsilon}$ and $\tau_{\text{biased}}^{\upsilon_{\max}} = \mu_{\delta_{\text{rand},x}}^{\upsilon} + 2\sigma_{\delta_{\text{rand},x}}^{\upsilon}$. Any metrics within these ranges could be due to chance with 95% probability.

We consider a dataset to be **biased** if models can perform well on them regardless of whether the embeddings used to train the model encode any information. Intuitively, these are datasets which are imbalanced in some way allowing the model to exploit this imbalance during prediction, but that do not reflect the knowledge encoded in the embeddings. To detect these, we look at model results for models trained on random embeddings (i.e. on models $m^{\delta_r, s_{rand}, t}$). We say that δ_r is biased if $\mu_{\delta_r, s_{rand}}^{f1}$ is outside of the $[\tau_{biased}^{f1}, \tau_{biased}^{f1}]$ range. The rationale is that even with random embeddings, such models were able to perform outside of the 95% baseline ranges.

We consider a trained model $m^{\delta_r,s,t}$ to be **significant** if its predictions are statistically better than predictions made by $m^{\delta_r,s_{\rm rand},t}$. Intuitively, this indicates that the embedding space *s* contributes information about relation *r* with high probability. Formally, we say that $m^{\delta_r,s,t}$ is significant if $\mu^{f_1}_{\delta_r,s} - \mu^{f_1}_{\delta_r,s_{\rm rand}} > 2\max(\sigma^{f_1}_{\delta_r,s}, \sigma^{f_1}_{\delta_r,s_{\rm rand}})$.

3 Measuring Lexico-Semantic Knowledge in Embeddings

We applied the method described above to two lexicosemantic KGs (WordNet and Sensigrafo) and several embedding spaces derived from different corpora and embedding algorithms. In this section we describe how we applied the methodology and present a summary of the results obtained. In the next section we discuss analyze the results.³

³The code we used is available at https://github.com/rdenaux/ embrelassess

	Sensigrafo	WordNet
version	14.2	3.0
words/lemmas	400K	155K
syn(set/con)	300K	118K
relations	55	27
derived rel datasets	149	27
pair types	lem2(lem,syn,POS)	lem2lem
	syn2(syn,POS)	

Table 1: Lexical Knowledge Graphs used.

3.1 Knowedge Graphs and Relations

We applied our methodology to two KGs: (i) WordNet, a well known lexico-semantic semantic network and (ii) Sensigrafo, a proprietary lexical knowledge graph developed by Expert System. Although similar in structure and scope –both KGs aim to provide a sense lexicon per language and use hypernymy as the main relation between senses–, Sensigrafo has been developed independently and is tightly coupled to Expert System's text analysis pipeline, enabling state of the art word sense disambiguation with claimed 90% accuracy.

Table 1 provides an overview and comparison for the KGs. WordNet does not have an explicit identifier for concepts; instead, it defines sets of synonyms, i.e. lemmas with a shared sense. Sensigrafo calls such sets syncons (synonymconcepts), since they refer to specific concepts and assigns a unique identifier to each. There are differences in terms of size, granularity of concepts, structure of the network (e.g. choice of central concepts) and types of relations. Finally, because Sensigrafo is developed and maintained as part of a text analytics pipeline, some of its features are tailored and biased towards supporting functionality and domains required by Expert System customers. By comparison, being a community effort WordNet may benefit from a wider set of stakeholders.

WordNet and Sensigrafo provide 27 and 55 types of relations respectively which we used to generate datasets. Table 2 shows an overview of the relations extracted; since discussing the 88 relations individually is unwieldy, we group the relations into types following –and expanding– a categorization described in the WordNet manual⁴. It suggests a top-level distinction between **Lexical** –those that hold between words– and **Semantic** –those that hold between words– relations also see, antonym, derivation, participle and pertainym as lexical. However, antonym is clearly based on the word's meaning. Derivation and participle seem to be truly lexical, while the others can also contain pairs which are semantically related, as shown by the example pairs in the table. Sensigrafo does not contain purely lexical relations.

The main relation type used by both KGs is **hypernymy**, which relates narrower to broader concepts (or instances to concepts). Although hypernymy is transitive, we only consider direct relations explicitly stated in the KG during the dataset generation.

Categorical relations are those that associate a concept with some category. When categories are concepts, they can be seen as an indirect hypernymy. WordNet has three types of categories called domains: *category* –concept to a topic domain–, *usage* –concepts to a type of use– and *region* – concepts to places. Sensigrafo has a set of core noun and verb concepts –called noun or verb categories, as well as tags– which form the backbone of the hypernymy hierarchy. These category syncons tend to be quite abstract concepts, hence we have only considered these relations at the syn2syn level, as they do not have lemmas in the seed vocabulary. Sensigrafo also defines a list of about 400 domains (which are not syncons), which is similar to WordNet's *category domain*.

Meronymy relates concepts in whole-member relations. WordNet distinguishes between *membership* –part of group–, *substance* –when something is made of substances that are orders of magnitude smaller than the whole– and *part* – remaining cases. Sensigrafo does not distinguish between these cases and uses this type of relation sparingly.

Synonymy can be defined between lemmas in the same synset/-con and is a special case of **conceptual similarity**. The *similarity* relation in WordNet captures pairs of similar adjectives. The *attribute* relation in WordNet relates adjectives describing a value for a noun (similar, but more limited than the *adjective-class* relation in Sensigrafo). The *cause* relation in WordNet describes causality between verbs and is similar in Sensigrafo. The *entailment* relation in WordNet describes an entailment between verbs, similarly to the *synconimplication* in Sensigrafo, which can also be applied to nonverbs. *Verb group* in WordNet groups similar verbs. *Synconunification* is assigned by linguists to syncon pairs that could be merged as a single syncon. *Antonym* describes concepts with opposite meanings.

Positional relations encode co-locations of concepts and are only provided by Sensigrafo. Two main subtypes: one relates adjectives or adverbs to other concepts while the second relates verbs to concepts that appear as the verb subject or object.

Sensigrafo provides about two dozens of **prepositional** relations between concepts. Such relations are of the type POS+preposition-POS. E.g. the pair *rival-titleholder* has relation *noun+to-noun*. Sensigrafo also encodes **geographic** relations between places. However, since the seed vocabulary did not contain many place names, we could not generate a dataset between lemmas.

Finally, we also generated datasets relating syncons to their **part-of-speech** (POS) besides the various $\delta_{random,x}$ as explained in Section 2 (not included in Table 2).

We generated a total of 176 datasets, based on the 88 relations and a seed vocabulary consisting of 76K concepts and 71K lemmas (roughly corresponding to our smallest embedding space, which was trained on a disambiguated version of the English United Nations corpus [Ziemski *et al.*, 2016]). For WordNet, we only generated datasets between lemmas. For Sensigrafo we also generated datasets between words, word/concepts and concepts.

3.2 Embeddings and Corpora

The word and concept embeddings studied were derived from 6 algorithms. Three provided embeddings based on sequences of either words or syncons: GloVe [Pennington *et al.*, 2014], FastText [Bojanowski *et al.*, 2017], Swivel [Shazeer *et*

⁴https://wordnet.princeton.edu/documentation/wninput5wn

type	name	KG	example pair	lem pairs	syn pairs	obj:subj	voc tot
Lexical	also see	W	mild-temperate	5800		1.25	2339
	derivation	W	revoke-revocation	118888		1.0	18094
	pertainym	W	regretfully-sorry	6516		1.37	6079
	participle of	W	operating-operate	81		2.13	94
Hypernymy	hypernym*	W	cinnamon-spice	110650		0.45	27048
	sup/subnomen	S	ditto	56413	33696	1.98	19768
	super/subverbum	s	kick-move	37660	9426	1.81	6630
	instance hypernym*	w	gemini-constellation	2358		0.44	1526
Categorical	category domain*	W	fly-air, tort-law	9116		0.13	4166
	sensiDomain	S	antitrust case-commercial law	28610		0.02	17636
	usage domain*	W	squeeze-slang	846		0.09	395
	region domain*	W	legionnaire-france	1349		0.12	546
	noun cat	S	flora-natural object		186797	0.004	50257
	verb cat	s	belong-v. of generic state		27388	0.01	11498
	tag	s	Christmas Eve-calendar day		31775	0.01	16825
Meronymy	member meronym*	W	archipelago-island	1315		1.51	1147
	substance meronym*	W	brine-sodium	369		0.98	378
	part meronym*	W	aeroplane-wing	6403		1.39	4054
	omni/parsnomen	S	construction-roofing	807	3110	1.22	597
Synonymy	synonym	w	encourage-promote	74822		1.0	22304
	synonym	S	ditto	69974		1.0	19130
Concept Simil	similar	W	big-immense	20464		1.0	6790
	attribute	w	short-length, good-quality	1718		1.0	859
	cause	W	secure-fasten	719		0.93	445
	syncon-cause	S	ring-sound, fright-fear	584		0.87	458
	entailment	W	look-see, peak-go up	1519		0.89	956
	syncon-implication	S	overtake-compete	1358	291	0.83	870
	verb group	W	shift-change, keep-prevent	4944		1.0	1193
	syncon-corpus	s	find-strike	97707	39644	1.11	10502
	syncon-unification	S	ritual-rite, enclose-envelop	6599	2392	0.99	3366
	antonym	W	release-detain	9310		1.0	4651
	(adj,n,v)antonym	s	ditto	4656	1728	1.03	1823
Positional	s-adjective-class	S	nightmarish-account	131853	33037	0.69	13972
	adverb-(n,v,adj,adv)	s	below-criteria	15598	3367	1.12	2509
	verb-object	s	counter-illness	108046	23163	1.24	9087
	verb-subject	s	less-tension, plunge-index	46949	10131	1.01	7126
Prepositional	12 internoun	S	meeting-colleague			0.83	10748
	10 verb prep noun	s	break down-tear			0.56	1848
	verb prep verb	s	set-rise	287		1.19	196
	4 adj prep noun	S	eligible-admission			1.62	299
Geographic	geography	S	Brussels-Brussels Capit. Reg.		1555	0.23	1584
Part-of-Speech	Noun	S	entity-GNoun		45961	n/a	76138
-	ProperNoun	S	Underground Railway-GPNoun		4134	n/a	8273
	Verb	s	cleanse-GVerb		11343	n/a	22691
	Adjective	S	record-GAdjective		12585	n/a	25175
	Adverb	s	in my opinion-GAdverb		2110	n/a	4225

Table 2: Overview of lexico-semantic relations studied.

al., 2016]. Two provided joint word and concept embeddings: Vecsigrafo [Denaux and Gómez-Pérez, 2017] based on a disambiguated corpus and HolE [Nickel *et al.*, 2016b], which directly generates embeddings from a KG (not from a corpus) and thus serves as a reference point. The embeddings were either publicly available or provided to us by [Denaux and Gómez-Pérez, 2017]. We also generated random embeddings s_{rand} for the seed vocabulary.

The embeddings were trained on three different corpora, which we chose to study whether relation prediction capacity varies depending on the corpus size: the English United Nations corpus[Ziemski *et al.*, 2016] (517M tokens), the English Wikipedia (just under 3B tokens) and Common Crawl (around 840B tokens). Although we aimed at using embeddings with 300 dimensions, in a few cases we had to diverge as the embeddings were only available with other dimensions.

3.3 Training and Results

We generated models based on fully connected neural networks (NN) with either 2 or 3 hidden layers. Initially we also generated models with logistic regression, which consistently underperformed. The 2 and 3 layer NNs generally produced similar results, suggesting they converge for the given datasets. For the standard case of embedding dimension 300, the input to the net is a vector of 600 dimensions, followed by hidden layers of 750 and 400 nodes for the NN2; and (750, 500, 250) for the NN3 (similar architectures were defined for single embedding relations such as syn2POS). The output layer has 2 nodes and uses 1-hot-encoding to encode positive or negative examples. For regularization, we use dropout between layers with value 0.5. We used a heuristic rule that varies the number of epochs to train a dataset depending on its size. Datasets with < 300 positive examples are trained for 48 epochs, those with < 5K for 24, < 30K for 12 and large datasets are trained for 6 epochs. We use the Adam optimizer with learning rate 1^{-5} and the cross entropy loss function. A scheduler reduces the learning rate on plateau. All of these hyper-parameters where derived through trial and error with a few sample relation datasets and kept constant to automatically train models without manual inspection. We used a random 90, 5, 5 split for training, validation and test from the input dataset. For WordNet relations we used a mixture of NN2 and NN3 models and trained each model either 3 or 5 times. As training the models is the main bottleneck of our approach, for Sensigrafo relations we only trained NN3 models, training each model 3 times.

We trained a total of 10,560 models, resulting in 1,596 evaluation metrics averaged over n runs: 126 for the random relation datasets, 149 trained on random embeddings, 1,065 for Sensigrafo relations and 268 for WordNet relations. Each run resulted in metrics for relation prediction on unseen pairs during training.

4 Analysis and Discussion of Results

Biased datasets and non-significant models We apply the prediction result analysis described in Section 2: based on 12 random datasets $\delta_{\text{rand},x}$ and 126 prediction results we obtained $\mu_{\delta_{\text{rand}}}^{\text{fl}} = 0.41$ and $\sigma_{\delta_{\text{rand}}}^{\text{fl}} = 0.12$, resulting in a base-

line range for prediction metrics (i.e. $[\tau_{biased}^{f1_{min}}, \tau_{biased}^{f1_{max}}])$ of [0.16, 0.65].

Using this range, we identify 38 of 156 datasets as being *biased*. With other words, even though we took care not to generate non-biased datasets by using negative switching, a little more than a quarter of the datasets generated in this way contains clues about the relation. Interestingly, word-pair datasets are much more likely to be biased (38%) while word/concept pairs are unlikely to so (only 7%) (see Table 3). These result suggest that using KGs as a *silver standard* [Paulheim, 2017] is of limited use for word-pair prediction, but is suitable when linking words to concepts.

We see that using a KG to build datasets for certain kind of relations is very difficult. This is the case for word pair datasets for categorical relations (all of the 52 generated datasets were biased); this is likely due to such relations being highly unbalanced with only a few words in the object position, further compounded by ambiguity of words compared to concepts. We think word ambiguity also plays a role in the difficulty producing non-biased datasets for positional relations.

On the bright side, about 71% of the models were trained on a non-biased dataset and we use these to identify (non)significant models. Overall, we found that 46.5% of the trained models using pre-trained embeddings were not statistically significant different from a baseline using random embeddings. About 1% of the models performed worse than the baseline. These were typically small datasets and relations that were hard to learn. In any case this 1% is below the 2.5% that can be expected by using the 2σ threshold for significance.

Conversely, only about 24% of the models trained using pre-trained embeddings significantly outperformed the baseline. For word/concept pair datasets this percentage jumps to almost 49% while for concept pairs it plummets to only 3.4%. Since most of the pre-trained embeddings we are using are corpus-based, this shows that such embeddings have trouble capturing the relations at the purely conceptual level. At the same time, these models are relatively good at relating words to concepts using semantic relations. In the sections below we only discuss results for the models that outperformed their baseline.

4.1 Relation Types in Embeddings

In the last two columns of Table 3 we see the absolute and relative f1 measures for the models that significantly outperformed the random baseline. We see that most of the results with an average f1 score higher than 0.8 are for relations between concepts; these were all achieved by training models on the HolE embeddings. Swivel embeddings also obtained good results for predicting categorical relation (0.846 f1, but this result just cleared the 2σ significance threshold). One model trained on FastText embeddings achieved (0.666 f1 on a meronymy relation). This confirms that *the studied corpusbased embeddings are not capable of capturing semantic relations at the concept level*.

For the models trained on word/concept or word pairs, we can go through the relation types. Absolute prediction accuracy for **lexical relations** is poor, the Vecsigrafo and GloVe

				models					metrics	
dataset rel	KG		datasets		non-predictable		"prec	lictable"	absolute	relative
		#	# biased	#	% biased	% not signif.	% better	% worse	$\mu_{ m f1}$	$\Delta \mu_{\rm f1}$
all	both	156	38	1281	28.6	46.5	23.9	0.9	0.687	0.173
all	wn	19	6	216	27.8	39.4	30.1	2.8	0.684	0.149
all	sensi	137	32	1065	28.8	48.	22.6	0.6	0.688	0.181
concent	sensi	44	10	293	22.9	737	34	0	0 870	0 608
word/concept	sensi	43	3	172	22.) 7.	43.6	48.8	0.6	0.664	0.000
word	sensi	50	19	600	38.	36.7	24.5	0.8	0.690	0.162
lexical	wn	4	0	32	0.	75.	25.	0.	0.652	0.167
hypernym	sensi	2	1	14	50	42.9	7.1	0	0.916	0.696
hypernym	sensi	$\frac{2}{2}$	1	8	50. 50	42.9	50	0.	0.699	0.070
hypernym _{w/c}	sensi	$\frac{2}{2}$	1	24	50.	0.	50. 50	0.	0.713	0.143
hypernym _w	wn	2	1	24	50.	4.2	37.5	8.3	0.756	0.106
categ.	sensi	3	1	21	33.3	52.4	14.3	0	0.891	0 671
categ	sensi	4	1	16	25	12.5	62.5	0.	0 744	0.263
categ	sensi	1	1	12	100	0	0	0.	0.711	0.205
categw	wn	4	4	40	100.	0.	0.	0.		
meronym	sensi	2	0	14	0	92.9	7.1	0	0.667	0.020
meronym _w	sensi	1	0	4	0.	50	50	0.	0.664	0.020
meronymw	sensi	1	Ő	12	0.	91.7	0.	8.3	0.001	0.290
meronym _w	wn	3	Ő	48	0.	66.7	31.2	2.1	0.708	0.166
synon.	sensi	0	0	0	0	0	0	0		
synon	sensi	1	0	4	0.	0. 25	75.	0.	0.804	0 249
synonw	sensi	1	Ő	12	0.	8.3	91.7	0.	0.677	0.114
synon _w	wn	1	Ő	8	0.	0.	100.	0.	0.680	0.135
simil.	sensi	6	1	42	16.7	81	2.4	0	0.909	0.688
similwe	sensi	7	0	28	0.	57.1	42.9	0.	0.624	0.210
similw	sensi	7	1	84	14.3	38.1	47.6	0.	0.628	0.128
simil _w	wn	5	1	64	12.5	43.8	39.1	4.7	0.655	0.153
position	sensi	6	1	42	16.7	78.6	4.8	0.	0.886	0.667
position	sensi	6	1	24	16.7	37.5	45.8	0.	0.721	0.120
position _w	sensi	7	5	84	71.4	21.4	6.	1.2	0.703	0.069
prepos _c	sensi	19	4	133	21.1	78.9	0.	0.		
prepos _{w/c}	sensi	22	0	88	0.	51.1	47.7	1.1	0.629	0.124
preposw	sensi	27	11	324	40.7	48.5	9.9	0.9	0.677	0.097
POS _c	sensi	5	1	20	20.	70.	10.	0.	0.882	0.665
POS _{w/c}	sensi	0	0	0	0.	0.	0.	0.		
POS_w	sensi	4	0	48	0.	2.1	97.9	0.	0.746	0.262

Table 3: Overview of results.

are the only embeddings that can predict pertainym relations with over 0.7 f1 score. Prediction of hypernym relations is good; the best performing model is trained on HolE on a word/concept dataset (0.797 f1); from the corpus-based embeddings, FastText performs best (0.766). Prediction of categorical relations (only for word/concept pairs) is quite good; corpus-based embeddings using Vecsigrafo perform similarly to HolE (around 0.82 f1) for nouns; corpus-based embeddings have trouble with categorical relations between verbs, with performance dropping below 0.6 compared to 0.8 for HolE. For meronymy relations, the datasets generated on Sensigrafo were rather small, resulting in f1 performances around 0.69 with Vecsigrafo embeddings; WordNet-derived datasets produce good performance with FastText, GloVe and Vecsigrafo having scores between 0.75 and 0.81 for the partmeronym relation; substance- and member-meronym relations accuracy drops below 0.7. Performance for synonym relations depends strongly on the used embedding; the best performers are HolE (0.93 f1), Vecsigrafo (0.79) and FastText (0.75) while Swivel performs poorly. Prediction of similarity relations is mediocre; the best embeddings are HolE (0.81), followed by GloVe and FastText (around 0.74); depending on the relation, prediction f1 can drop to 0.6. For positional relations, Vecsigrafo (0.76) outperforms even HolE (0.74) with FastText and Glove trailing (0.7); this is similar for prepositional relations. For part-of-speech, FastText and GloVe perform well (f1 above 0.8).

4.2 Impact of embedding type and corpus size

Unsurprisingly, the larger the corpus the better the overall results for Sensigrafo datasets. Average f1 scores for embeddings trained on the UN corpus, Wikipedia and Common Crawl were 0.66, 0.69 and 0.73. For WordNet datasets (only lem2lem), the scores are similar: 0.68, 0.72 and 0.71. Thus, although increasing the corpus size helps, for many types of relations, the gain from training on a very large corpus is relatively small.

Table 4 summarizes the performance of the different embedding learning algorithms for predicting relations grouped by the pair type. For word pair prediction, FastText outperforms other embedding learning algorithms, including HolE (although this difference is not major). HolE excels at predicting relations between senses, but its performance decreases as lemmas are introduced, since it cannot disambiguate between the senses. Vecsigrafo and GloVe both are not far behind the performance of FastText and HolE, but produce significant predictions for more relations than Fast-Text and HolE. Standard Swivel with words lags behind, especially in the number of relations that it can predict.

Impact of joint word-concept learning

Vecsigrafo co-trains word and concept embeddings. Table 4 shows that compared to Swivel, this co-training improves, for word pairs, both the number of relations that can be predicted (double the number) and the average f1 score (we assume that the additional predicted relations push the average score down). Similarly, we see that compared to word pairs, Vecsigrafo is able to produce predictions for more relations when considering word/concept pairs. As discussed above, corpus-

datasets	algo	F1 _{avg}	F1 _{std}	pair type
8	HolE	0.90	0.02	concept
1	Swivel	0.85	0.0	concept
1	FastText	0.67	0.0	concept
26	HolE	0.67	0.09	word/concept
48	Vecsigrafo	0.64	0.08	word/concept
38	FastText	0.74	0.08	word
12	HolE	0.72	0.09	word
43	Vecsigrafo	0.68	0.07	word
43	GloVe	0.68	0.08	word
21	Swivel	0.66	0.06	word

Table 4: Average F1 scores for predicting relations with different pair types and embeddings.

based, joint word-concept training does not seem to capture relations between at the concept level, suggesting there is room to improve such algorithms.

5 Conclusion

This paper presented a methodology for studying whether embeddings capture relations as well as KGs and applied it to study lexico-semantic relations between words and concepts. The results show that for a few relations, word embeddings can outperform embeddings derived directly from KGs. Also, corpus-based embeddings fail to capture relations at the concept level. However, for most relation types, embeddings only can predict relations with an accuracy under 0.7. Our results provide evidence that correct capture of relations should happen at the concept level and may not be achievable with high accuracy at the word level. As future work, we want to apply our method to study contextual embeddings.

Acknowledgments

The research reported in this paper is supported by the EU Horizon 2020 programme, under grants European Language Grid-825627 and Co-inform-770302.

References

- [Bahdanau *et al.*, 2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. sep 2014.
- [Bojanowski et al., 2017] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching Word Vectors with Subword Information. *Transactions of the Association of Computational Linguistics*, 5(1):135–146, 2017.
- [Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating Embeddings for Modeling Multi-relational Data, 2013.
- [Cho *et al.*, 2014] Kyunghyun Cho, Bart Van Merriënboer, and Dzmitry Bahdanau. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. pages 103–111, 2014.
- [Denaux and Gómez-Pérez, 2017] Ronald Denaux and Jose Manuel Gómez-Pérez. Towards a Vecsigrafo:

Portable Semantics in Knowledge-based Text Analytics. In International Workshop on Hybrid Statistical Semantic Understanding and Emerging Semantics @ISWC17, 2017.

- [Denaux and Gomez-Perez, 2019] Ronald Denaux and Jose Manuel Gomez-Perez. Vecsigrafo: Corpus-based word-concept embeddings. *Semantic Web*, (Preprint):1– 28, 2019.
- [Faruqui et al., 2015] Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. Retrofitting Word Vectors to Semantic Lexicons. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1606–1615, Stroudsburg, PA, USA, 2015. Association for Computational Linguistics.
- [Fu et al., 2014] Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. Learning Semantic Hierarchies via Word Embeddings. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1199– 1209, 2014.
- [Gábor et al., 2017] Kata Gábor, Haifa Zargayouna, Isabelle Tellier, Davide Buscaldi, and Thierry Charnois. Exploring Vector Spaces for Semantic Relations. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1815–1824, 2017.
- [Garcia and Gomez-Perez, 2018] Andres Garcia and Jose Manuel Gomez-Perez. Not just about size - A Study on the Role of Distributed Word Representations in the Analysis of Scientific Publications. apr 2018.
- [Kalchbrenner and Blunsom, 2013] Nal Kalchbrenner and Phil Blunsom. Recurrent Continuous Translation Models. pages 1700–1709, 2013.
- [Khot *et al.*,] Tushar Khot, Ashish Sabharwal, and Peter Clark. SCITAIL: A Textual Entailment Dataset from Science Question Answering.
- [Kim, 2014] Yoon Kim. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Stroudsburg, PA, USA, 2014. Association for Computational Linguistics.
- [Levy et al., 2015] Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. Do Supervised Distributional Methods Really Learn Lexical Inference Relations? Naacl-2015, pages 970–976, 2015.
- [Li *et al.*,] Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. Visualizing and Understanding Neural Models in NLP. pages 681–691.
- [Lin et al., 2016] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Neural Relation Extraction with Selective Attention over Instances. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2124–2133, 2016.

- [Melo and Paulheim, 2017] André Melo and Heiko Paulheim. Detection of Relation Assertion Errors in Knowledge Graphs. In *kcap*, 2017.
- [Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality, 2013.
- [Nickel et al., 2016a] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. Proceedings of the IEEE, 104(1):11–33, 2016.
- [Nickel *et al.*, 2016b] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic Embeddings of Knowledge Graphs. *AAAI*, pages 1955–1961, oct 2016.
- [Parikh,] Ankur P Parikh. A Decomposable Attention Model for Natural Language Inference. pages 2249–2255.
- [Paulheim, 2017] Heiko Paulheim. Knowledge Graph Refinement: A Survey of Approaches and Evaluation Methods. *Semantic Web*, 8(3):489–508, 2017.
- [Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.
- [Riedel et al., 2013] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. Relation Extraction with Matrix Factorization and Universal Schemas. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, number June, pages 74– 84, 2013.
- [Ristoski and Paulheim, 2016] Petar Ristoski and Heiko Paulheim. RDF2Vec: RDF Graph Embeddings for Data Mining. pages 498–514. Springer, Cham, oct 2016.
- [Roller et al., 2014] Stephen Roller, Katrin Erk, and Gemma Boleda. Inclusive yet Selective: Supervised Distributional Hypernymy Detection. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, August 23-29 2014, pages 1025–1036, 2014.
- [Schnabel et al., 2015] Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. Evaluation methods for unsupervised word embeddings. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 298–307, 2015.
- [Seo *et al.*, 2016] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hananneh Hajishirzi. BI-DIRECTIONAL ATTENTION FLOW FOR MACHINE COMPREHEN-SION. *arXiv preprint*, 2016.
- [Shazeer *et al.*, 2016] Noam Shazeer, Ryan Doherty, Colin Evans, and Chris Waterson. Swivel: Improving Embeddings by Noticing What's Missing. *arXiv*, feb 2016.
- [Shi and Weninger, 2017] Baoxu Shi and Tim Weninger. ProjE: Embedding Projection for Knowledge Graph Completion. *Thirty-First AAAI Conference on Artificial Intelli*gence, February 2017.

- [Shwartz and Dagan, 2016] Vered Shwartz and Ido Dagan. Path-based vs. Distributional Information in Recognizing Lexical Semantic Relations. In *COLING*, 2016.
- [Sutskever *et al.*,] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to Sequence Learning with Neural Networks.
- [Turney and Mohammad, 2015] Peter D. Turney and Saif M. Mohammad. Experiments with Three Approaches to Recognizing Lexical Entailment. *Natural Language Engineering*, 21(3):437–476, 2015.
- [Zeiler and Fergus, 2014] Matthew D. Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. pages 818–833. Springer, Cham, 2014.
- [Ziemski *et al.*, 2016] Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. The united nations parallel corpus v1. 0. In *Language Resource and Evaluation*, 2016.