Semantic Web 0 (0) 1
IOS Press

Vecsigrafo: Corpus-based Word-Concept Embeddings

Bridging the Statistic/Symbolic Representational Gap

Ronald Denaux [a,*],
Jose Manuel Gomez-Perez [a,**]
[a] *Cogito Labs, Expert System, Madrid, Spain*
E-mails: rdenaux@expertsystem.com, jmgomez@expertsystem.com

Abstract. The proliferation of knowledge graphs and recent advances in Artificial Intelligence have raised great expectations related to the combination of symbolic and distributional semantics in cognitive tasks. This is particularly the case of knowledge-based approaches to natural language processing as near-human symbolic understanding and explanation rely on expressive structured knowledge representations that tend to be labor-intensive, brittle and biased. This paper reports research addressing such limitations by capturing as embeddings in a joint space both words and concepts from large document corpora. We compare the quality of the resulting embeddings and show that they outperform word-only embeddings for a given corpus.

Keywords: concept embeddings, word embeddings, learning embeddings, corpus-based embeddings, knowledge graphs

1. Introduction

The history of Artificial Intelligence is a quest for the perfect balance between expressivity and sheer reasoning power. Early AI systems developed during the '70s showed that reasoning capabilities alone do not suffice if the system does not work at a level of expressivity that can be understood by humans. Powerful rule-based systems [1] failed because their reasoning formalism was focused on the operational aspects of inference. Lacking abstraction and proper explanation, the gap between the knowledge representation formalism and domain experts became unbridgeable, resulting in what was known ever after as the knowledge acquisition bottleneck[2]. In an attempt to address this challenge and work at the knowledge level rather than the operational one [3], the knowledge acquisition task became a modeling activity instead of a task consisting of eliciting knowledge from the mind of the expert.

Along the knowledge level path came ontologies, semantic networks and knowledge graphs. Among other fields in AI, knowledge graphs endow natural language processing (NLP) with rich, expressive and actionable descriptions of the domain of interest and support logical explanations of reasoning outcomes. On the downside, they can be costly to produce since they require a considerable amount of human effort to manually encode knowledge in the required representation formalisms, which can also be excessively rigid and brittle, and subject to human bias. Furthermore, knowledge graphs are typically built top-down, without taking into account, or leveraging, the wealth of available data.

In parallel, the last decade has witnessed a dramatic shift towards statistical methods to text understanding due to the increasing availability of raw data and cheaper computing power. Such methods have proved to be powerful and convenient in many linguistic tasks. Particularly, recent results in the field of distributional semantics have shown promising ways to learn language models from text, encoding the meaning of each word in the corpus as a vector in dense, low-dimensional spaces. Among their applications, word embeddings have proved to be useful in term simi-

*Corresponding author. E-mail: rdenaux@expertsystem.com.
**Corresponding author. E-mail: jmgomez@expertsystem.com.

larity, analogy and relatedness, as well as many NLP downstream tasks.

As a matter of fact, word embeddings are usually at the input layer of deep learning architectures for NLP, including e.g. classification or machine translation. However, the proper extraction of meaning from the text is left to the neural net and to a large extent depend on the size and variety of the training corpora. Additionally, although there is interesting progress in areas like computer vision [4], intermediate layers are generally hard to interpret or match to entities and concepts[5], the typical nodes in knowledge graphs.

Many argue [6–8] that knowledge graphs can enhance both expressivity and reasoning power in statistical approaches to NLP and advocate for a hybrid approach leveraging the best of both worlds. This is particularly the case in situations where there is not enough data or adequate methodology to learn the nuances associated with the concepts and their relationships, which on the other hand can be explicitly represented in a knowledge graph. However, the application of knowledge graphs to produce disambiguated joint word and concept embeddings following a hybrid knowledge formalism involving statistic and symbolic representations is still largely unexplored. Moreover, leveraging statistical, corpus-based methods to capture tacit knowledge and extend symbolic representations in a knowledge graph, alleviating brittleness, also remains a challenge.

In this paper we focus on the above-mentioned challenges and discuss our corpus-based, joint word-concept algorithm, while studying how the resulting embeddings compare to existing word, knowledge graph, and hybrid embeddings. We run a comprehensive set of experiments with different learning algorithms over a selection of corpora in varying sizes and forms and evaluate our results over a variety of tasks, both intrinsic (semantic similarity, relatedness) and extrinsic (word-concept and hypernym prediction). In doing so, we also propose a number of mechanisms to measure the quality and properties of the resulting embeddings, including word and concept prediction plots and inter-embedding agreement. Our results show that our approach consistently outperforms word-only and knowledge graph embeddings and most of the hybrid baselines with a medium size training corpora, remaining on a par against other systems using much larger corpora.

The paper is structured as follows. Next section provides an overview of the research context relevant to our work in areas including word, graph and sense embedding. Section 3 describes our approach to capture as embeddings the semantics of both words and concepts in large document corpora. Section 4 goes on to evaluate our results over different datasets and tasks, comparing to the approaches described in section 2. Next, section 5 reflects on our findings and provides a deep insight and interpretation of the evaluation results. Finally, section 6 concludes the paper and advances next steps and applications of our research.

2. Related Work

To the best of our knowledge, this is the first work that studies jointly learning embeddings for words and concepts from a large disambiguated corpus. The idea itself is not novel, as [9] points out, but performing a practical study is difficult due to the lack of manually sense-annotated datasets. The largest such dataset is SemCor [10] (version 3.0), a corpus of 537K tokens, 274K of which are annotated with senses from WordNet. Although this dataset could be used with our approach to generate embeddings for the WordNet senses, results from work on word-embeddings show that the size of the corpus greatly affect the quality of the learned embeddings and that corpora in the order of billion tokens are required. In this paper we use an automatic approach for generating word-concept annotations, which makes it possible to use large corpora to learn good quality concept and word embeddings as our studies and results in section 4 show.

Although no directly related approaches have been proposed, we discuss several other approaches varying from those for learning plain word-embeddings, to those learning sense and concept embeddings from corpora and semantic networks, and those which do not use corpora at all, but instead attempt to learn concept embeddings directly from a knowledge graph.

2.1. Word Embeddings

Learning word embeddings[1] has a relatively long history [11], with earlier works focused on deriving embeddings from co-occurrence matrices and more recent work focusing on training models to predict words based on their context[12]. Both approaches are roughly equivalent as long as design choices and hyperparameter optimization are taken into account[13].

[1] Also called the Vector Space Model in the literature.

Most of the recent work in this area was triggered by the Word2Vec algorithm proposed in [14] which provided an efficient way to learn word embeddings by predicting words based on their context words and using negative sampling. Recent improvements on this family of algorithms[15] also take into account (i) subword information by learning embeddings for 3 to 6 character n-grams, (ii) multi-words by pre-processing the corpus and combining n-grams of words with high mutual information like "New_York_City" and (iii) learning a weighting scheme (rather than pre-defininig it) to give more weight to context words depending on their relative position to the target word. These advances are available via the FastText implementation and pre-calculated embeddings.

Algorithms based on word co-occurrences are also available. GloVe [16] and Swivel[17] are two algorithms which learn embeddings directly from a sparse co-occurrence matrix that can be derived from a corpus.

These approaches have been shown to learn lexical and semantic relations. However, since they stay at the level of words, they suffer from issues regarding word ambiguity. And since most words are polysemic, the learned embeddings must either try to capture the meaning of the different senses or encode only the meaning of the most frequent sense. In the opposite direction, the resulting embedding space only provides an embedding for each word, which makes it difficult to derive an embedding for the concept based on the various words which can be used to refer to that concept.

The approach described in this paper is an extension that can be applied to both word2vec style algorithms and to co-occurrence algorithms. In this paper we only applied this extension to Swivel, although applying it to GloVe and the standard word2vec implementations should be straightforward. Applying it to FastText would be more complicated, especially when taking into account the sub-word information, since words can be subdivided into char n-grams, but concepts cannot.

2.2. Sense and Concept Embeddings

A few approaches have been proposed to produce sense and concept embeddings from corpora. One approach to resolve this is to generate *sense embeddings* [18], whereby the corpus is disambiguated using Babelfy and then word2vec is applied over the disambiguated version of the corpus. Since plain word2vec is applied, only vectors for senses are generated. Jointly learning both words and senses was proposed by [19] and [20] via multi-step approaches where the system first learns word embeddings, then applies disambiguation based on WordNet and then learns the joint embeddings. While this addresses ambiguity of individual words, the resulting embeddings do not directly provide embeddings for KG-concepts, only to various synonymous word-sense pairs[2].

Another approach for learning embeddings for concepts based on a corpus without requiring word-sense disambiguation is NASARI[9], which uses lexical specifity to learn concept embeddings from wikipedia subcorpora. These embeddings have as their dimensions, the lexical specificity of words in the subcorpus, hence they are sparse and harder to apply than low-dimensional embeddings such as those produced by word2vec. For this reason, NASARI also proposes to generate "embedded vectors" which are weighted averaged vectors from a conventional word2vec embedding space. This approach only works for wikipedia and BabelNet, since you need a way to create a subcorpus that is relevant to entities in the knowledge base. Furthermore, this approach only seems to produce embeddings for nouns.

Finally, the work that is closest to our work is SW2V (Senses and Words to Vectors) [21] which proposes a lightweight word-disambiguation algorithm and extends the Continuous Bag of Words architecture of word2vec to take into account both words and senses. Our approach is essentially the same, although there are various implementational differences: (i) we use our proprietary disambiguator; (ii) implemented our learning algorithm as a variation of correlation-based algorithms as a consequence (iii) we take into account the distance of context words and concepts to the target word. In terms of evaluation, [21] only reports results for 2 word-similarity datasets while we provide an extensive analysis on 14 datasets. We further analyse the impact of different corpus sizes and look into the inter-agreement between different vector spaces.

2.2.1. Graph Embeddings

Several approaches have been proposed to create concept embeddings directly from knowledge graphs, such as TransE[22], HolE[23], ProjE[24], RDF2Vec[25] and Graph Convolutions[26]. The main

[2] E.g. word-sense pairs $\texttt{apple}^{\texttt{N}}_{2}$ and $\texttt{Malus\_pumila}^{\texttt{N}}_{1}$ have separate embeddings, but the concept for *apple tree* they represent has no embedding.

goal of such concept embeddings is typically graph completion. In our opinion, these approaches all have the same drawback: they encode the knowledge (including biases) explicitly contained in the source knowledge graph, which is typically already a condensed and filtered version of the real world data. Even large knowledge graphs only provide a fraction of the data that can be gleaned from raw datasets such as wikipedia and other web-based corpora. I.e. these embeddings cannot learn from raw data as it appears in the real-world. In our evaluation we have used HolE to compare how such word and concept embeddings compare to those derived from large text corpora.

3. Corpus-Based Joint Concept-Word Embeddings

In order to build hybrid systems which can use both bottom-up (corpus-based) embeddings and top-down (KG) knowledge, we propose to generate embeddings which share the same vocabulary as the Knowledge Graphs. This means generating embeddings for knowledge items represented in the KG such as concepts and surface forms (words and expressions) associated to the concepts in the KG[3].

The overall process for learning joint word and concept embeddings is depicted in Figure 1, we start with a text corpus on which we apply tokenization and word sense disambiguation (WSD) to generate a *disambiguated corpus*, which is a sequence of **lexical entries** (words, or multiword expressions). Some of the lexical entries are annotated with a particular sense (concept) formalised in the KG. To generate embeddings for both senses and lexical entries, we need to correctly handle lexical entries which are associated to a sense in the KG, hence we extend the matrix construction phase of the Swivel [17] algorithm to generate a co-occurrence matrix which includes both lexical forms and senses as part of the vocabulary as explained below. Then we apply the training phase of a slightly modified version of the Swivel algorithm to learn the embeddings for the vocabulary; the modification is the addition of a vector regularization term as suggested in [27] (equation 5) which aims to reduce the distance between the column and row (i.e. focus and context) vectors for all vocabulary elements.

Modified Co-occurrence Matrix Construction The main modification from standard Swivel[4] is that in our case, each token in the corpus is not a single word, but a lexical entry with an optional KG-concept annotation. Both lexical entries and KG-concepts need to be taken into account when calculating the co-occurrence matrix. Formally, the co-occurence matrix $X \in \mathbb{R}^{V*V}$ contains the co-occurrence counts found over a corpus, where $V \subset L \cup C$ is the vocabulary, which is a conjunction of lexical forms L and KG-concepts C. $X_{ij} = \#(v_i, v_j)$ is the frequency of lexical entries or concepts v_i and v_j co-occurring within a certain window size w. Note that $X_{ij} \in \mathbb{R}$, since this enables us to use a dynamic context window [13], weighting the co-occurrence of tokens according to their distance within the sequences.[5]

4. Evaluation

Our approach requires a few changes to conventional algorithms for learning word embeddings, in particular the tokenization and lemmatization required to perform disambiguation affects the vocabulary. Furthermore, the introduction of concepts in the same vector space can affect the quality of word embeddings. Obviously the whole point of such a hybrid approach is to be able to learn both high quality word and concept embeddings. Hence, we posit the following research questions:

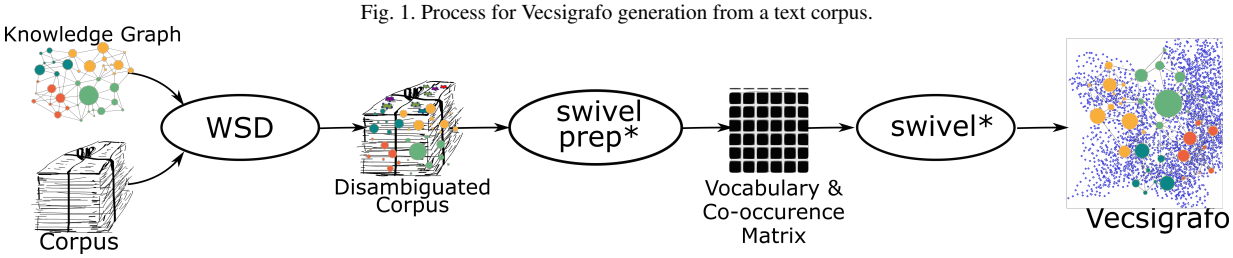
- How does vecsigrafo compare to conventional word embeddings? More specifically:
  - Does inclusion of concepts in the same space affect the quality of the word embeddings?
  - we know that corpus size affects the quality of word embeddings, does this effect change for vecsigrafo-based embeddings?
  - How does vecsigrafo (based on Swivel) compare to other word-embedding algorithms

[3] In RDF, this typically means values for `rdfs:label` properties, or words and expressions encoded as `ontolex:LexicalEntry` instances using the lexicon model for ontologies (see https://www.w3.org/2016/05/ontolex/).

[4] As implemented in https://github.com/tensorflow/models/tree/master/research/swivel

[5] We use a modified harmonic function $h(n) = 1/n$ for $n > 0$ and $h(0) = 1$ which covers the case where a token has both a lexical form and a concept. This is the same weighing function used in GloVe and Swivel; word2vec uses a slightly different function $d(n) = n/w$.

Fig. 1. Process for Vecsigrafo generation from a text corpus.



– How do corpus-based derived embeddings compare to other concept-embeddings such as KG-derived embeddings and lexical specificity derived embeddings?

In an attempt to find answers to the research questions, we study the resulting embeddings using a few tasks that provide an indication about their quality:

– word-similarity. We analyse results for 14 word-similarity datasets for word and concept relatedness. Besides testing on the embedding agreement with human-labeled gold standards, we also check inter-agreement between embeddings generated via different methods, which is a good indicator that the resulting embeddings are converging. Inter-agreement also provides evidence about how much the resulting word embeddings learned in vecsigrafo differ from conventional word embeddings.
– word-prediction. We use a test corpus to simulate how well the resulting embeddings predict a word, based on its context words. This essentially recreates the word2vec loss function, but uses a test corpus (unseen during training). This task provides insight into both the quality of the resulting embeddings. Also, since this task provides information about a subset of the vocabulary it can be used to generate plots which provide an overview of possible disparities in the quality of common and uncommon words.
– hypernym prediction. While word-similarity and word-prediction tasks are intrinsic evaluations, ultimately the goal of learning hybrid concept embeddings is to be able to refine knowledge representations. One such refinement is the prediction of specific relations in a knowledge graph. We study whether different embeddings can be used to predict hypernym relations between words.

4.1. Corpora, Knowledge Graphs and Embeddings

In our experiments, we use both pre-calculated word embeddings, but for better comparison we have tried to generate embeddings using available code and the same input corpus whenever possible. In this section, we first describe the corpora we have used as well as those third parties have reported using for generating pre-calculated embeddings. Then, we also describe how we have generated embeddings, including relevant metadata and training parameters.

Table 1 provides an overview of the corpora used for generating embeddings. To study the effect of the corpus size (and domain of the input corpus), we have used the United Nations corpus[28] as an example of a medium sized corpus that is domain specific. This corpus consists of transcriptions of sessions at the United Nations, hence the contents are domain specific with topics in politics, economics and diplomacy being predominant. We have used the English part of the corpus that is aligned with Spanish[6]. As an example of a larger corpus, we have used the dump of the English Wikipedia from January 2018. Embeddings provided by third parties include the UMBC corpus[30], a web-corpus of roughly the same dimensions as the Wikipedia corpus. To compare our embeddings to those trained on a very large corpus, we use pre-calculated GloVe embeddings that were trained on CommonCrawl[7].

Besides the text corpora, the tested embeddings contain references to concepts defined in two knowledge graphs. Our proprietary semantic network is call Sensigrafo, we have used the vanilla English Sensigrafo (released with Cogito Studio 14.2[8]), which contains around 400K lemmas and 300K concepts. Sensigrafo is similar to WordNet, it is the result of person-decades

[6]Cross-lingual applications of the embeddings is not in the scope of this paper, although we discuss some initial applications in [29]
[7]http://commoncrawl.org
[8]http://www.expertsystem.com/products/cogito-cognitive-technology

Table 1
Evaluation corpora

Corpus	tokens	unique	freq
UNv1.0 en-es en	517M	2.7M	469K
wiki-en-20180120	2.89B	49M	5M
UMBC	2.95B		
CommonCrawl	840B		

of continuous curation by a team of linguists. Like WordNet, it the core relation between concepts is that of hypernymy, but various other lexical and semantic relations are also included. Another difference with WordNet is that Sensigrafo has explicit identifiers for concepts, while WordNet has no such identifiers, instead WordNet uses a set of synonyms (each of which is a word sense) which refer to the same concept. The second semantic network we use in our experiments is BabelNet 3.0, which has about 14M concepts (7 million of which are named entities). We have not trained embeddings on top of BabelNet, athough we have included BabelNet derived embeddings[9, 21] in our studies.

Table 2 shows an overview of the embeddings used during the evaluations. We used five main methods to generate these. In general we tried to use embeddings with 300 dimensions, although in some cases we had to deviate.

- Vecsigrafo based embeddings were first tokenized and word-disambiguated using Cogito. We explored two basic tokenization variants, first is lemma-concept with filtered tokens ("ls filtered"), whereby we only keep lemmas and concept ids for the corpus. Lemmatization uses the known lemmas in Sensigrafo to combine compound words as a single token. The filtering step removes various types of words: dates, numbers, punctuation marks, articles, proper names (entities), auxiliary verbs, proper nouns and pronouns which are not bound to a concept. The main idea of this filtering step is to remove tokens from the corpus which are not semantically relevant. We also trained a few embeddings without lemmatization and filtering. In such cases, we have kept the original surface form bound to the concept (including morphological variants) and we did not remove the tokens described above. For all the embeddings, we have used a minimum frequency of 5 and a window size of 5 words around the target word. We also used a harmonic weighting scheme (we experimented with linear and and uniform weighting schemes but results did not differ substantially).
- Swivel[9] based embeddings using either a basic white-space tokenization of the input corpus, or a lemma-based tokenization performed by Cogito. We have used the default parameters defined by the open-source project. For the wikipedia corpus, we had to reduce the number of dimensions to 256, since otherwise, the main Swivel algorithm would run out of GPU memory during training. We also imposed a limit of 1M for the vocabulary for the same reason.
- GloVe embeddings trained by us were derived using the master branch on its GitHub repository[10] and we used the default hyper-parameters defined therein.
- FastText embeddings trained by us were derived using the master branch on its GitHub repository[11] and we used the default hyper-parameters defined therein.
- HolE embeddings were trained by us using the code on GitHub[12] after we exported the Sensigrafo to create a training set of 2.5M triples including covering over 800K lemmas and syncons and 93 relations, including hypernymy relations, but also `hasLemma` relations between concepts and lemmas (We also tried to apply ProjE[13], but various errors and slow performance made it impossible to apply it to our Sensigrafo corpus.). We trained HolE for 500 epochs using 150 dimensions and the default hyper-parameters. The final evaluation after training reported an MRR of 0.13, a mean rank of 85279 and Hits10 of 19.48%.

[9] https://github.com/tensorflow/models/tree/master/research/swivel
[10] https://github.com/stanfordnlp/GloVe
[11] https://github.com/facebookresearch/fastText/commit/3872afadb3a9f30de7c7792ff2ff1bda64242097
[12] https://github.com/mnick/holographic-embeddings/commit/c2db6e1554e671ab8e6acace78ec1fd91d6a4b90
[13] https://github.com/bxshi/ProjE

Besides the embeddings trained by us, we also include, as part of our study, several pre-calculated embeddings, notably the GloVe embeddings for CommonCrawl –code `glove_840B` provided by Stanford[14]–, FastText embeddings based on a wikipedia dump from 2017 –code `ft_en`[15], as well as the embeddings for BabelNet concepts (NASARI and SW2V) since these require direct access to BabelNet indices. In Table 2 we report the details that are reported by the embedding providers.

4.2. Word Similarity

Word similarity tasks are one of the most common intrinsic evaluations that are used to evaluate the quality of embeddings[12]. Although there are issues with these types of tasks[31, 32], they tend to provide insights into how well learned embeddings capture the perceived semantic relatedness between words. One of our hypotheses is that introducing concepts to the vector space should help to learn embeddings which better capture word similarities; hence this type of evaluation should prove useful. Furthermore, it is possible to extend the default word-similarity task –whereby the cosine similarity between the vectors of a pair of words is compared to a human-rated similarity measure– by calculating a concept-based similarity measure: in this case, we select the maximum similarity between the concepts associated to the initial pair of words. This intuitively makes sense since, presumably, when a pair of words is related, human raters naturally disambiguate the senses that are closest rather than taking into account all the possible senses of the words. We first describe the 14 word similarity datasets that we are using in our evaluation and then present the results.

4.2.1. Word-similarity Datasets

The RG-65 dataset [33] was the first one generated in order to test the distributional hypothesis. Although it only has 65 pairs, the human ratings are the average of 51 raters. MC-30 [34] is a subset of RG-65, which we include in our studies in order to facilitate comparison with other embedding methods. The pairs are mostly nouns.

Another classic word similarity dataset is WS-353-ALL [35] which contains 353 word pairs. 153 of these were rated by 13 human raters and the remaining 200 by 16 subjects. The pairs are mostly nouns, but also include some proper names (people, organizations, days of the week). Since the dataset mixes similarity and relatedness, [36] used WordNet to split the dataset into a WS-353-REL and WS-353-SIM containing 252 and 203 word pairs respectively (some unrelated word pairs are included in both subsets).

YP-130 [37] was the first dataset focusing on pairs of verbs. The 130 pairs were rated by 6 human subjects. Another dataset for verbs is VERB-143 [38] which contains verbs in different conjugated forms (gerunds, third person singular, etc.) rated by 10 human subjects. The most comprehensive dataset for verbs is SIMVERB3500 [39] consisting of 3500 pairs of verbs (all of which are lemmatized), they were rated via crowdsourcing by 843 raters and each pair was rated by at least 10 subjects (over 65K individual ratings).

MTurk-287 [40] is another crowdsourced dataset focusing on word and entity relatedness. The 287 word pairs include plurals and proper nouns and each pair was rated on average by 23 workers. MTurk-771 [41] also focuses on word relatedness and was crowdsourced with an average of 20 ratings per word pair. It contains pairs of nouns and rare words were not included in this dataset.

MEN-TR-3K[42] is another crowd-sourced dataset which combines word similarity and relatedness. As opposed to previous datasets, where raters gave an explicit score for pair similarity, in this case raters had to make comparative judgements between two pairs. Each pair was rated against 50 other pairs by the workers. The dataset contains mostly nouns (about 81%), but also includes adjectives (about 13%) and verbs (about 7%), where a single pair can mix nouns and adjectives or verbs. The selected words do not include rare words.

SIMLEX-999 [43] is a crowd-sourced dataset that explicitly focuses on word similarity and contains (non-mixed) pairs of nouns (666), adjectives (111) and verbs (222). This dataset also provides a score of the level of abstractness of the words. Since raters were explicitly asked about similarity and not relatedness, pairs of related –but not similar– words, receive a low score. The 500 raters each rated 119 pairs and each pair was rated by around 50 subjects.

RW-STANFORD [44] is a dataset that focuses on rare (infrequent) words. Words still appear in WordNet (to ensure they are English words as opposed to foreign words). Each of the 2034 pairs was rated by 10 crowd-sourced workers. The dataset contains a mix

[14] http://nlp.stanford.edu/data/glove.840B.300d.zip

[15] https://s3-us-west-1.amazonaws.com/fasttext-vectors/wiki.en.vec

Table 2
Evaluated embeddings.

Code	Corpus	Method	Tokenization	Epochs	Vocab	Concepts
	UN	vecsi	ls filtered	80	147K	76K
	UN	swivel	ws	8	467K	0
	UN	glove	?	15	541K	0
	UN	vecsi	ts	8	401K	83K
	UN	fastText	?	15	541K	0
	wiki	glove	?	25	2.4M	0
	wiki	swivel	ws	8	1.0M	0
	wiki	vecsi	ls filtered	10	824K	209K
`ft_en`	wiki	fastText	?	8	2.4M	0
	UMBC	w2v	?	?	1.3M	0
	wiki/UMBC	nasari	?	?	5.7M	4.4M
	sensigrafo	HolE	n/a	500	825K	423K
	wiki'	fastText	?	?	2.5M	0
`glove_cc`	CommonCrawl	GloVe	?	?	2.2M	0

of nouns (many of which are plurals), verbs (including conjugated forms) and adjectives.

Finally, SEMEVAL17 (English part of task 2) [45] provides 500 word pairs, selected to include named entities, multi-words and to cover different domains. They were rated in such a way that different types of relations (synonymy, similarity, relatedness, topical association and unrelatedness) align to the scoring scale. The gold-standard similarity score was provided by three annotators.

4.2.2. Results

Table 3 shows the Spearman correlation scores for the 14 word similarity datasets and the various embeddings generated based on the UN corpus. The last column in the table shows the average coverage of the pairs for each dataset. Since the UN corpus is medium sized and focused on specific domains, many words are not included in the learned embeddings, and hence, the scores are only calculated based on a subset of the pairs.

Table 4 shows the results for the embeddings trained on larger corpora and directly on the sensigrafo. We have not included results for vectors trained with NASARI (concept-based), word2vec and sw2v on UMBC, since these perform consistently worse than the remaining embeddings.

Table 5 shows the aggregate results. Since some of the word similarity datasets overlap —SIMLEX-999 and WS-353-ALL were split into its subsets, MC-30 is a subset of RG-65— and other datasets —RW-STANFORD, SEMEVAL17, VERB-143 and MTURK-287— have non-lemmatised words (plurals and conjugated verb forms) which penalise embeddings that use some form of lemmatisation during tokenisation, we take the average Spearman score over the remaining datasets.

4.3. Inter-embedding Agreement

The word similarity datasets are typically used to assess the correlation between the similarity of word pairs assigned by embeddings and a gold standard defined by human annotators. However, we can also use the word similarity datasets to assess how similar two embedding spaces are. We do this by collecting all the similarity scores predicted for all the pairs in the various datasets and calculating the Spearman's ρ metric between the various embedding spaces. We present the results in Figure 2.

4.4. Word-Concept Prediction

One of the disadvantages of word similarity (and relatedness) datasets is that they only provide a single metric per dataset. In [29] we introduced Word-prediction plots, a way to visualise the quality of embeddings by performing a task that is very similar to the loss objective of word2vec. Given a test corpus (ideally different from the corpus used to train the embeddings), iterate through the sequence of tokens using a context window. For each focus word, take the (weighted) average of the embeddings for the context tokens and compare it to the embedding for the focus word using cosine similarity. If the cosine similarity is close to 1, this essentially correctly predicts the tar-

Table 3

Spearman correlations for word similarity datasets and **UN**-based embeddings. The column names refer to the method used to train the embeddings, the tokenization of the corpus (**l**emma, **s**yncon and or **t**ext and whether the tokens were **f**iltered), and whether **c**oncept-based word similarity was used instead of the usual word-based similarity.

dataset	ft	glove	swivel	swivel l f	vecsi ls f	vecsi ls f c	vecsi ts	vecsi ts c	avg_{perc}
MC-30	0.602	0.431	0.531	0.572	0.527	0.405	0.481	**0.684**	82.5
MEN-TR-3k	0.535	0.383	0.509	0.603	**0.642**	0.525	0.558	0.562	82.0
MTurk-287	0.607	0.438	0.519	0.559	**0.608**	0.578	0.500	0.540	69.3
MTurk-771	0.473	0.398	0.416	0.539	**0.599**	0.497	0.520	0.520	94.6
RG-65	0.502	0.378	0.443	0.585	0.614	0.441	0.515	**0.664**	74.6
RW-STANFORD	0.492	0.263	0.356	0.444	**0.503**	0.439	0.419	0.353	49.2
SEMEVAL17	0.541	0.395	0.490	0.595	**0.635**	0.508	0.573	0.610	63.0
SIMLEX-999	0.308	0.253	0.226	0.303	**0.382**	0.349	0.288	0.369	96.1
SIMLEX-999-Adj	0.532	0.267	0.307	0.490	**0.601**	0.559	0.490	0.532	96.6
SIMLEX-999-Nou	0.286	0.272	0.258	0.337	**0.394**	0.325	0.292	0.384	94.7
SIMLEX-999-Ver	0.253	0.193	0.109	0.186	0.287	**0.288**	0.196	0.219	100.0
SIMVERB3500	0.233	0.164	0.155	0.231	0.306	**0.328**	0.197	0.318	94.4
VERB-143	**0.382**	0.226	0.116	0.162	0.085	-0.089	0.234	0.019	76.2
WS-353-ALL	0.545	0.468	0.516	0.537	**0.588**	0.404	0.502	0.532	91.9
WS-353-REL	0.469	0.434	0.465	0.478	**0.516**	0.359	0.447	0.469	93.4
WS-353-SIM	0.656	0.553	0.629	0.642	**0.699**	0.454	0.619	0.617	91.5
YP-130	0.432	0.350	0.383	0.456	**0.546**	0.514	0.402	0.521	96.7

Table 4

Spearman correlations for word similarity datasets on large corpora (UMBC, wikipedia and CommonCrawl).

corpus	sensi		umbc	wiki17	wiki18				cc	
dataset	HolE	HolE c	sw2v c	ft en	ft	glove	swivel	vecsi ls f	vecsi ls f c	glove	avg_{perc}
MC-30	0.655	**0.825**	0.822	0.812	0.798	0.565	0.768	0.776	0.814	0.786	100.0
MEN-TR-3k	0.410	0.641	0.731	0.764	0.760	0.607	0.717	**0.785**	0.773	**0.802**	99.9
MTurk-287	0.272	0.534	0.633	0.679	0.651	0.473	**0.687**	0.675	0.634	**0.693**	85.6
MTurk-771	0.434	0.577	0.583	0.669	0.649	0.504	0.587	**0.685**	0.578	**0.715**	99.9
RG-65	0.589	0.798	0.771	0.797	0.770	0.639	0.733	0.803	**0.836**	0.762	100.0
RW-STANFORD	0.216	0.256	0.395	0.487	**0.492**	0.124	0.393	0.463	0.399	0.462	81.9
SEMEVAL17	0.475	0.655	**0.753**	0.719	0.728	0.546	0.683	0.723	0.692	0.711	81.8
SIMLEX-999	0.310	0.380	**0.488**	0.380	0.368	0.268	0.278	0.374	0.420	0.408	99.4
SIMLEX-999-Adj	0.246	0.201	0.556	0.508	0.523	0.380	0.323	0.488	**0.564**	**0.622**	99.5
SIMLEX-999-Nou	0.403	0.484	**0.493**	0.410	0.383	0.321	0.331	0.422	0.464	0.428	100.0
SIMLEX-999-Ver	0.063	0.133	**0.416**	0.231	0.233	0.105	0.103	0.219	0.163	0.196	97.7
SIMVERB3500	0.227	0.318	**0.417**	0.258	0.288	0.131	0.182	0.271	0.331	0.283	98.8
VERB-143	0.131	-0.074	-0.084	0.397	**0.452**	0.228	0.335	0.207	0.133	0.341	75.0
WS-353-ALL	0.380	0.643	0.597	0.732	**0.743**	0.493	0.692	0.708	0.685	0.738	98.5
WS-353-REL	0.258	0.539	0.445	0.668	**0.702**	0.407	0.652	0.649	0.609	0.688	98.2
WS-353-SIM	0.504	0.726	0.748	0.782	**0.805**	0.615	0.765	0.775	0.767	0.803	99.1
YP-130	0.315	0.550	**0.736**	0.533	0.562	0.334	0.422	0.610	0.661	0.571	98.3

get word based on its context. By aggregating all such cosine similarities for all tokens in the corpus we can (i) plot the average cosine similarity for each term in the vocabulary that appears in the test corpus and (ii) get an overall score for the test corpus by calculating the (weighted by token frequency) average over all the words in the vocabulary.

Table 6 provides an overview of the test corpora we have chosen to generate word and concept prediction scores and plots. The corpora are:

Table 5
Aggregated word similarity results.

method	corpus	avg ρ	avg coverage %
glove	cc	0.629	100.0
vecsi ls f c 25e	wiki	0.622	99.6
vecsi ls f 25e	wiki	0.619	98.6
sw2v c	umbc	0.615	99.9
ft 8e	wiki	0.613	100.0
vecsi ls f c 10e	wiki	0.609	99.6
ft	wiki17	0.606	98.9
HolE c 500e	sensi	0.566	99.6
w2v	umbc	0.566	98.9
swivel 8e	wiki	0.542	99.9
vecsi ls f 80e	UN	0.538	93.1
vecsi ts c 8e	UN	0.505	97.9
swivel l f	UN	0.480	92.9
ft 15e	UN	0.451	88.6
vecsi ts 8e	UN	0.443	91.0
glove 25e	wiki	0.438	100.0
vecsi ls f c 80e	UN	0.433	83.4
swivel 8e	UN	0.403	87.9
HolE 500e	sensi	0.381	99.6
glove_{15e}	UN	0.364	88.6
nasari c	umbc	0.360	94.0
sw2v	umbc	0.125	100.0

– webtext [46] is a corpus of contemporary text fragments (support fora, movie scripts, ads) scraped from publicly accessible websites that has been used as training data for many NLP applications. The corpus was created by downloading web pages to create a topic-diverse collection.
– NLTK gutenberg selections[16] contains a sample of public-domain (hence originally published at least 80 years ago) literary texts by well-known authors (Shakespeare, Jane Austen, Walt Whitman, etc.) from Project Gutenberg.
– europarl-10k. We have created a test dataset based on the Europarl [47] v7 dataset. We used the English file that has been parallelised with Spanish, removed the empty lines and kept only the first 10K lines. We expect Europarl to be relatively similar to the UN corpus since they both provide transcriptions of proceedings in similar domains.

Figure 3 shows the word prediction plots for various embeddings and the three test corpora. Table 7 shows (i) the token coverage relative to the embedding vocabulary (i.e. the percentage of the embedding vocabulary found in the tokenised test corpus); (ii) the weighted average score, this is the average cosine similarity per prediction made (however, since frequent words are predicted more often, this may skew the overall result if infrequent words have worse predictions.); (iii) the "token average" score, this is the average of the average score per token. This gives an indication of how likely you are to predict a token (word or concept) given its context if you select a token from the embedding vocabulary at random (i.e. without taking into account its frequency in general texts).

[16] https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/packages/corpora/gutenberg.zip

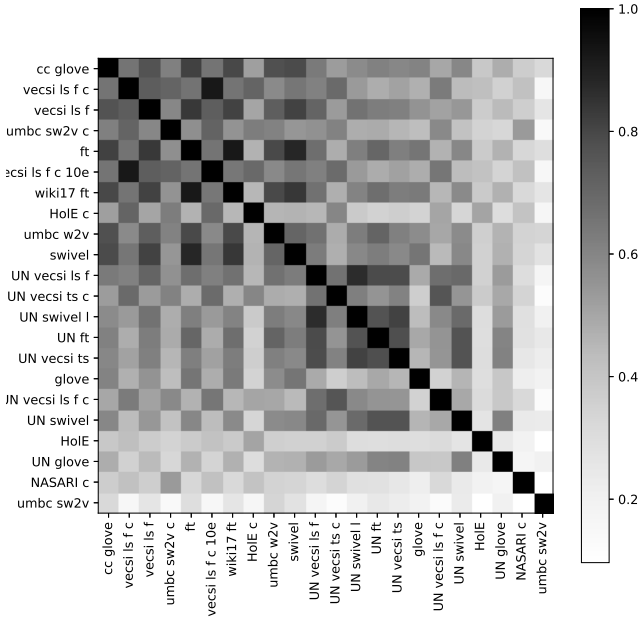


Fig. 2. Inter-embedding agreement for the word similarity datasets in the same order as Table 5. Embeddings that do not mention a corpus, were trained on Wikipedia 2018.

Table 6
Overview - test corpora used to gather word and concept prediction data.

	tokens		
corpus	text	lemmas	concepts
webtext	300K	209K	198K
gutenberg	1.2M	868K	832K
europarl-10k	255K	148K	143K

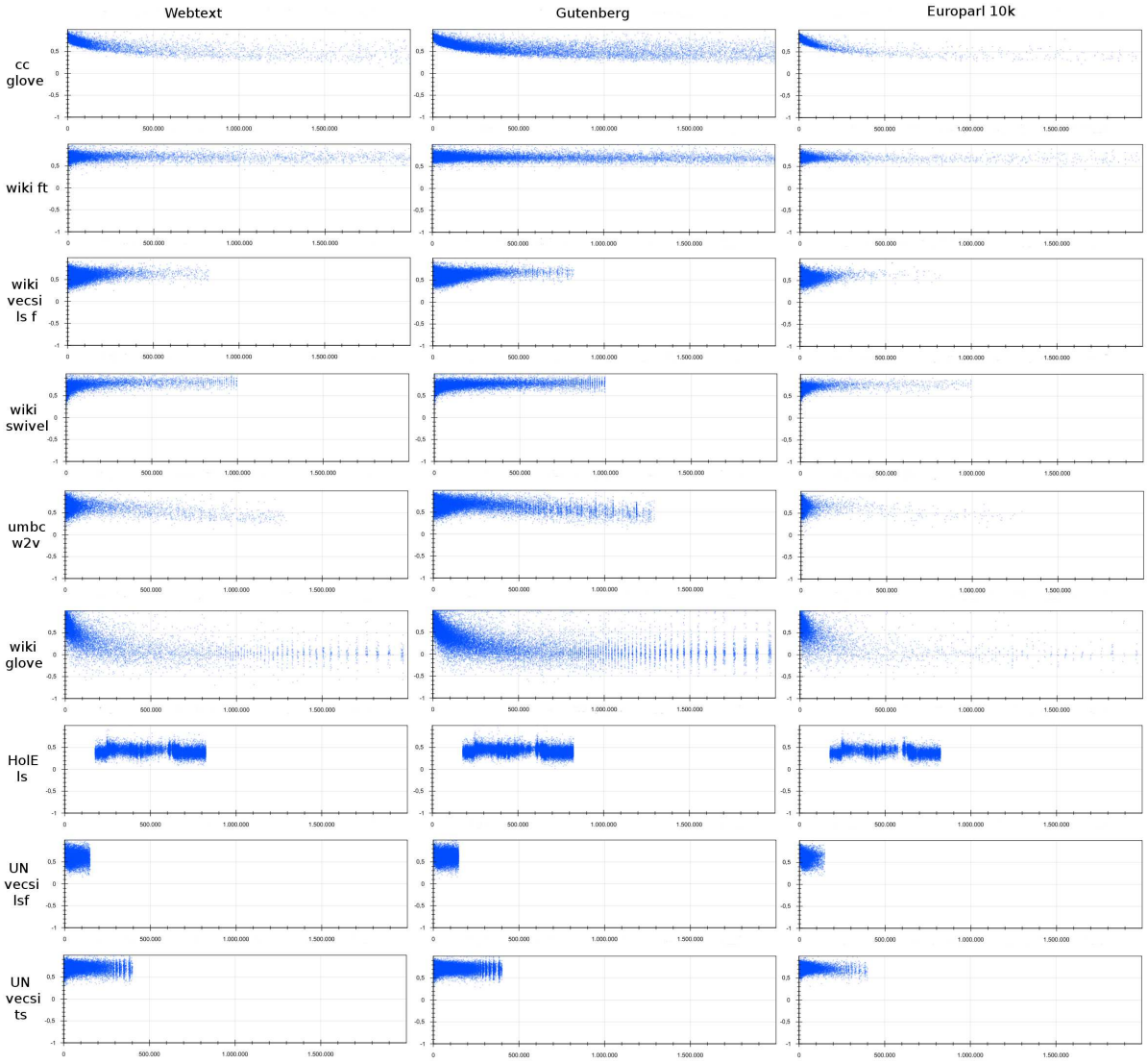


Fig. 3. Word and Concept prediction plots. The horizontal axis contains the word ids sorted by frequency on the training corpus; although different embeddings have different vocabulary sizes, we have fixed the plotted vocabulary size to 2M tokens to facilitate comparison. Since HolE is not trained on a corpus, hence the frequencies are unknown, the vocabulary is sorted alphabetically. The vertical axis contains the average cosine similarity between the weighted context vector and the target word or concept.

4.5. Relation prediction

Word (and concept) similarity and prediction tasks are good for getting a sense of the embedding quality. However, ultimately the relevant quality metric for embeddings is whether they can be used to improve the performance of deep learning systems that perform more complex tasks such as document categorization or knowledge graph completion. For this reason we include an evaluation for predicting specific types of relations in a knowledge graph between pairs of words. At Expert System, such a system would help our team of linguists to curate the Sensigrafo.

Table 7

Aggregate word prediction values. The coverage refers to the percentage of tokens (words and concepts) in the embedding vocabulary that were found in the test corpus. The "w avg" is the average cosim weighted by token frequency and "t avg" is the average cosine similarity for all the token predictions regardless of their frequency in the corpus.

test corpus	webtext			gutenberg			Europarl-10k		
emb	coverage	w avg	t avg	coverage	w avg	t avg	coverage	w avg	t avg
cc glove	0.007	**0.855**	**0.742**	0.016	**0.859**	0.684	0.005	**0.868**	**0.764**
wiki swivel	0.013	0.657	**0.703**	0.027	0.664	**0.718**	0.010	0.654	0.666
UN vecsi ts	0.069	**0.688**	**0.703**	0.103	0.701	0.715	0.062	0.700	**0.717**
wiki ft	0.006	0.684	0.702	0.013	**0.702**	0.712	0.004	**0.702**	0.700
umbc w2v	0.012	0.592	0.638	0.030	0.574	0.662	0.008	0.566	0.649
UN vecsi ls f	0.138	0.630	0.617	0.214	0.652	0.628	0.128	0.681	0.636
wiki vecsi ls	0.037	0.603	0.593	0.057	0.606	0.604	0.026	0.601	0.588
HolE ls	0.035	0.414	0.416	0.056	0.424	0.424	0.026	0.400	0.398
wiki glove	0.006	0.515	0.474	0.013	0.483	0.408	0.004	0.468	0.566

To minimise introducing bias, rather than using Sensigrafo as our knowledge graph, we have chosen to use WordNet since we have not used it to train HolE embeddings and it is different from Sensigrafo (hence any knowledge used during disambiguation should not affect the results). For this experiment, we chose relations

- `verb group` which relates similar verbs to each other, e.g "shift"-"change" and "keep"-"prevent".
- `entailment` which describes entailment relations between verbs, e.g. "peak"-"go up" and "tally"-"count".

Datasets We built a dataset for each relation by (i) starting with the vocabulary of `UN vecsi ls f` (the smallest vocabulary for the embeddings we are studying) and look up all the synsets in WordNet for the lemmas. Then we (ii) searched for all the connections to other synsets using the selected relations, which gives us a list of positive examples. Finally, (iii) we generate negative pairs by generating pairs based on the list of positive examples for the same relation (this *negative switching* strategy has been recommended in order to avoid models simply memorising words associated to positive pairs[**?**]. This resulted in a dataset of 3039 entailment pairs (1519 positive) and 9889 verb group pairs (4944 positive).

Training Next, we trained a neural net with 2 fully-connected hidden layers on each dataset, using a 90 % training, 5 validation, 5 test split. The neural nets received as their input the concatenated embeddings for the input pairs (if the input verb was a multi-word like "go up", we took the average embedding of the constituent words when using word embeddings rather than lemma embeddings). Therefore, for embeddings with 300 dimensions, the input layer had 600 nodes, while the two hidden layers had 750 and 400 nodes. The output node has 2 one-hot-encoded nodes. For the HolE embeddings, the input layer had 300 nodes and the hidden layers had 400 and 150 nodes. We used dropout (0.5) between the hidden nodes and an Adam optimizer to train the models for 12 epochs on the verb group dataset and 24 epochs on the entailment dataset. Also, to further avoid the neural net to memorise particular words, we include a random embedding perturbation factor, which we add to each input embedding; the idea is that the model should learn to categorise the input based on the difference between the pair of word embeddings. Since different embedding spaces have different values, the perturbation takes into account the minimum and maximum values of the original embeddings.

Figure 4 shows the results of training various of the embeddings: `cc glove`, `wiki ft`, `HolE`, `UN vecsi ls f` and `wiki vecsi ls f`. Since constructing such datasets is not straightforward [13], we also include a set of random embeddings. The idea is that, if the dataset is well constructed, models trained with the random embeddings should have an accuracy of 0.5, since no relational information should be encoded in the random embeddings (as opposed to the trained embeddings).

5. Discussion

Based on the data gathered and presented in the previous section, we now revisit our research questions and discuss the results.

5.1. Vecsigrafo (and sw2v) compared to conventional word embeddings

From tables 3 and 5 we can draw the conclusion that, for the UN corpus (a medium sized corpus):

- co-training lemmas and concepts produces better embeddings than training them using conventional word embedding methods. In particular we see that: $\rho_{vecsi_{lsf}} > \rho_{swivel_l} \simeq \rho_{ft} \succ \rho_{vecsi_{ts}} \succ \rho_{swivel} \simeq \rho_{glove}$ Where $>$ means that the difference is statistically significant (t-test $p < 0.01$), $\succ$ means slightly significance ($p < 0.05$) and $\simeq$ means difference is not statistically significant. We see that for the same tokenization strategy (**l**emmas with **f**iltering or plain **t**ext), adding concepts significantly improves the quality of the word embeddings. Furthemore, we see that just lemmatizing and filtering achieves a similar quality as that of FastText (which also performs preprocessing and uses sub-word information as discussed in section 2.1).
- concept-based word similarity suggests concept embeddings are better if co-trained with plain text tokenization ($\rho_{ts_c} \succ \rho_{ts}$), but worse when co-trained with filtered lemmas ($\rho_{lsf} \succ \rho_{lsf_c}$).

For larger corpora such as the wikipedia and UMBC:

- there is no statistically significant difference between using FastText, Vecsigrafo (either concept-based or lemma-based similarity) or SW2V (concept-based). Similarly, GloVe performs at roughly the same level as these other embeddings but requires a very large corpus such as CommonCrawl to match them.
- Standard Swivel and GloVe perform significantly worse than FastText, Vecsigrafo and SW2V.
- For Vecsigrafo based embeddings, both lemma and concept embeddings are of high quality. For SW2V-based embeddings, concept embeddings are of high quality, but the co-trained word embeddings are of poor quality. Since both methods are similar, it is not clear why this is the case.
- NASARI concept embeddings (based on lexical specificity) are of poor quality compared to other embeddings. This was unexpected, since results in [9] were very good for similar word-similarity tests (although restricted to a few datasets). Maybe this is due to the fact that these embeddings only take into account concepts which are nouns, but even for noun-based datasets we could not reproduce the results reported in [9]: for MC-30 we measured 0.68 ρ vs 0.78 reported, for SIMLEX-999-Nou we measured 0.38 instead of 0.46 and WS-353-SIM it was 0.61 instead of 0.68.

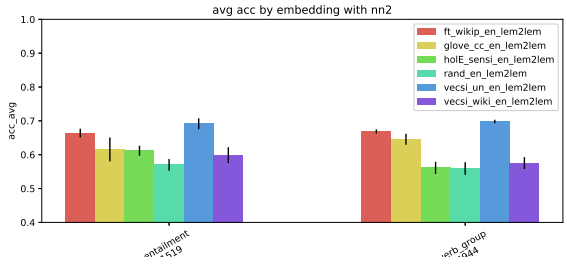


Fig. 4. Entailment and Verb Group average prediction accuracy over 5 training runs. The black bars show the standard deviation.

In terms of inter-embedding agreement, from figure 2 we see that concept-based embeddings tend to have a higher agreement with other concept-based embeddings, even if those concepts are derived from a different semantic net (BabelNet and Sensigrafo). Similarly, word and lemma based embeddings tend to have a higher inter-agreement with other word-based embeddings. Since both types of embeddings achieve high scores for word-similarity (against the gold standard), this suggests that a hybrid approach could yield better results.

Furthermore, we clearly see that for the medium sized corpus, all embeddings tend to have a high inter-agreement. For larger corpora, this difference in corpus is not as important as the method used to train the embeddings (vecsigrafo, fastText, ws2v, etc.) or the method used to predict word similarity (word-based vs concept-based)

From the word prediction plots (figure 3) and results (table 7, we see very different learning patterns for the various word embedding algorithms:

- GloVe tends to produce skewed predictions excelling at predicting very high-frequency words (with little variance), but as words become less frequent the average prediction accuracy drops and variance increases. This patterns is particularly clear for GloVe trained on Common Crawl. The same pattern seems to apply for `wiki glove`, however, the plot shows that for most words (except the most frequent ones) these embeddings barely perform better than random (average cosine similarity is close to 0). This suggests that there is an issue with the default hyper

parameters, or that GloVe requires a much higher number of epochs compared to other algorithms (note we initially trained most of the embeddings with 8 epochs, but due to poor performance we increased the presented GloVe embeddings for wikipedia to 25 epochs).
- FastText produces very consistent results: prediction quality does not seem to change depending on the word frequency
- word2vec applied to UMBC seems to have a pattern in between that of FastText and GloVe. It shows a high variance in prediction results, especially for very high-frequency words and shows a linearly declining performance as words become less frequent.
- Swivel with standard tokenization also shows mostly consistent predictions; however very frequent words show a higher variance in prediction quality which is almost the opposite of GloVe: some high-frequency words tend to have a poor prediction score, but the average score for less frequent words tends to be higher. The same pattern seems to apply to Vecsigrafo (based on swivel), although it is less clear for `wiki vecsi ls`. Due to the relatively small vocabulary sizes for the studied vecsigrafos trained on the UN corpus, it is hard to make out a learning pattern when normalising the vocabulary to 2M words.

By comparing the word-prediction results between `wiki swivel` and the three vecsigrafo-based embeddings we can see a few counter-intuitive results.

- First, on average word prediction quality *decreases* by using vecsigrafo with lemmatisation and filtering, which is surprising (especially since word embedding quality seems to improve significantly based on the word-similarity results as discussed above). One possible reason for this is that the context vector for vecsigrafo-based predictions will typically be the average of twice as many context tokens (since it will include both lemmas and concepts). However, the results for `UN vecsi ts` would suffer from the same issue, but this does not seem to be the case. In fact, `UN vecsi ts` performs as well as `wiki swivel` at this task.
- Second, both UN-based vecsigrafo embeddings outperform the wiki-based vecsigrafo embedding for this task. When comparing `UN vecsi ls f` and `wiki vecsi ls`, we see that due to the vocabulary size, the UN-based embeddings had to perform fewer predictions for fewer tokens; hence maybe less frequent words are introducing noise when performing word prediction. Further studies are needed in order to explain these results. For now, the results seem to indicate that, for word-prediction task, vecsigrafo embeddings based on smaller corpora outperform those trained on larger corpora. This is especially relevant for tasks such as vecsigrafo based disambiguation, for which standard word embeddings would not be useful.

Other results from the word-prediction study are:

- most embeddings seem to perform better for the gutenberg test corpus than for webtext. The only exceptions are `cc glove` and `wiki glove`. This may be a result of the size of the test corpus (gutenberg is an order or magnitude larger than webtext) or the formality of the language. We assume that webtext contains more informal language, which is not represented in either Wikipedia or the UN corpus, but could be represented in CommonCrawl. Since the average differences are quite small, we would have to perform further studies to validate these new hypotheses.
- the training and test corpora matter: for most embeddings we see that the token average for Europarl is similar or worse than for webtext (and hence worse than for Gutenberg). However, this does not hold for the embeddings that were trained on the UN corpus, which we expect to have a similar language and vocabulary as Europarl. For these embeddings –`UN vecsi ts` and `Un vecsi ls f`– the Europarl predictions are better than for the Gutenberg dataset. Here again, the GloVe-based embeddings do not conform to this pattern. Since the `wiki glove` embeddings are of poor quality, this is not that surprising. For `cc glove`, it is unclear why results would be better than for both webtext and gutenberg.
- Finally and unsurprisingly, lemmatization clearly has a *compacting effect on the vocabulary size*. This effect can provide practical advantages: for example, instead of having to search for the top-k neighbours in a vocabulary of 2.5M words, we can limit our search to 600K lemmas (and avoid finding many morphological variants for the same word).

From the verb relation prediction results in figure 4, we see that, once again, `UN vecsi ls f` outperforms other embeddings, including `wiki vecsi ls f`. The fact that the random embeddings result in an average accuracy of around 0.55 indicates that the dataset are well formed and the results are indicative of how well the trained models would perform for new pairs of words. We can see that both tasks are relatively challenging, with the models performing at most at around 70% accuracy.

5.2. Vecsigrafo compared to KG embeddings

Table 5 shows that for KG-based embeddings, the lemma embeddings (`HolE 500e`) perform poorly, while the concept-based similarity embeddings perform relatively well (`HolE c 500e`). However, the concept embeddings learned using HolE perform significantly worse than those based on the top-performing word embedding methods (FastText on wiki and GloVe on CommonCrawl) and concept-embedding methods (sw2v and vecsigrafo). This seems to validate our hypothesis that corpus-based concept-embeddings can improve on graph-based embeddings since they can refine the concept representations by taking into account tacit knowledge from the training corpus, which is not explicitly captured in a knowledge graph. In particular, and unsurprisingly, lemma embeddings derived from KGs are of much poorer quality as those derived from (disambiguated) text corpora.

The inter-embedding agreement results from figure 2 show that HolE embeddings have a relatively low agreement with other embeddings, especially conventional word-embeddings. Concept-based HolE similarity results have a relatively high agreement with other concept-based similarities (vecsigrafo, sw2v and NASARI).

Results from the word-prediction task are consistent with those of the word-similarity task. HolE embeddings perform poorly when applied to predicting a target word or concept from context tokens.

In Figure 3 we see that the first 175K words in the HolE vocabulary are not represented in the corpus. The reason for this is that these are quoted words or words referring to entities (hence capitalized names for places, people) which have been filtered out due to the `ls f` tokenization applied to the test corpus. Also, we see a jump in token prediction quality around word 245K which is maintained until word 670K. This corresponds to the band of concept tokens, which are encoded as `en#concept-id`. Hence words between 175K and 245K are lemmas starting from "a" to "en" and words after 670K are lemmas from "en" to "z". This again indicates that HolE is better at learning embeddings for concepts rather than lemmas (leaf nodes in the Sensigrafo KG).

6. Conclusions and Future Work

In this paper we presented Vecsigrafo, a novel approach to produce corpus-based, joint word-concept embeddings from large disambiguated corpora. Vecsigrafo brings together statistical and symbolic knowledge representations in a single, unified formalism for NLP. Our results over 14 datasets and different intrinsic and extrinsic tasks (word similarity, word prediction and hypernym prediction) show that our approach consistently outperforms word-only, graph and other hybrid embedding approaches with a medium size training corpora, leveling out in the presence of much larger corpora.

Word embeddings have shown to learn lexical and semantic relations but, staying at the level of words, they suffer from word ambiguity and brittleness when it comes to capture the different senses in a word. As a consequence, these methods usually require very large amounts of training text. Previous lemmatization and word-sense disambiguation of the training corpora enables Vecsigrafo to capture each sense much more efficiently, requiring considerably smaller corpora while producing higher quality embeddings. In the case of graph embeddings, these approaches are limited to the knowledge explicitly described in the knowledge graph, which is just a condensed interpretation of the domain according to a knowledge engineer. Vecsigrafo, on the other hand, learns from the way language is expressed in the real world and uses this knowledge to complement and extend the knowledge graph. Finally, compared to previous sense and concept embeddings, Vecsigrafo explicitly provides embeddings for knowledge graph concepts, can be used with different knowledge graphs, and covers not only nouns but also all the lexical entries that are semantically relevant.

In this paper we have also proposed two mechanisms that have proved useful to provide a deeper insight on the quality of the resulting embeddings. Word prediction plots allow overcoming the main limitation of word similarity (and relatedness) benchmarks, which only provide a single metric per dataset, by using the embeddings to predict a word based on its

context in three additional test corpora. On the other hand, inter-embedding agreement use the word similarity datasets to assess how similar two embedding spaces are.

Future research directions will seek to enrich, validate and extend the coverage of existing knowledge graphs as well as to continue our work in cross-lingual, cross-modal scenarios (see, e.g. [48]). We will also deepen in the understanding of the interplay between corpus size and the quality of the resulting vecsigrafo embeddings. For instance, the hypernym prediction task has shown that it is not always the case that Vecsigrafo will obtain better results when the size of the training corpus is increased. A possible explanation could be that small and medium sized corpora benefit from the added semantics and expansion coming from the knowledge graph, but when the corpus is significantly large and the different lexical forms associated to lemmas and concepts become statistically more significant, the improvement curve for the resulting embeddings becomes less steep. Finally, at Expert System we are applying Vecsigrafo to optimize Cogito in various ways, including assisted extension and curation of the underlying knowledge graph, cross-lingual support over (currently supported) 14 languages or enhanced disambiguation.

Acknowledgements

We gratefully acknowledge the EU H2020 program for grant DANTE-700367 and project GRESLADIX-IDI-20160805 funded by CDTI (Spain).

References

[1] S.U.H.P. Project and E.H. Shortliffe, *MYCIN: a Knowledge-based Computer Program Applied to Infectious Diseases*, 1977. https://books.google.es/books?id=JpfCHwAACAAJ.

[2] E.A. Feigenbaum, The Art of Artificial Intelligence: I. Themes and Case Studies of Knowledge Engineering, Technical Report, Stanford, CA, USA, 1977.

[3] A. Newell, The Knowledge Level, *Artif. Intell.* **18**(1) (1982), 87–127, ISSN 0004-3702. doi:10.1016/0004-3702(82)90012-1. http://dx.doi.org/10.1016/0004-3702(82)90012-1.

[4] M.D. Zeiler and R. Fergus, Visualizing and Understanding Convolutional Networks, in: *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele and T. Tuytelaars, eds, Springer International Publishing, Cham, 2014, pp. 818–833. ISBN 978-3-319-10590-1.

[5] C. Olah, A. Mordvintsev and L. Schubert, Feature Visualization, *Distill* (2017), https://distill.pub/2017/feature-visualization. doi:10.23915/distill.00007.

[6] A. Sheth, S. Perera, S. Wijeratne and K. Thirunarayan, Knowledge Will Propel Machine Understanding of Content: Extrapolating from Current Examples, in: *Proceedings of the International Conference on Web Intelligence*, WI '17, ACM, New York, NY, USA, 2017, pp. 1–9. ISBN 978-1-4503-4951-2. doi:10.1145/3106426.3109448. http://doi.acm.org/10.1145/3106426.3109448.

[7] Y. Shoham, Why Knowledge Representation Matters, *Commun. ACM* **59**(1) (2015), 47–49, ISSN 0001-0782. doi:10.1145/2803170. http://doi.acm.org/10.1145/2803170.

[8] P. Domingos, A Few Useful Things to Know About Machine Learning, *Commun. ACM* **55**(10) (2012), 78–87, ISSN 0001-0782. doi:10.1145/2347736.2347755. http://doi.acm.org/10.1145/2347736.2347755.

[9] J. Camacho-Collados, M.T. Pilehvar and R. Navigli, NASARI: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities, *Artificial Intelligence* **240** (2016), 36–64, ISSN 00043702. ISBN 0004-3702. doi:10.1016/j.artint.2016.07.005. www.elsevier.com/locate/artint.

[10] G.A. Miller, C. Leacock, R. Tengi and R.T. Bunker, A Semantic Concordance, *Proceedings of the Workshop on Human Language Technology - HLT '93* (1993), 303–308. ISBN 1558603247. doi:10.3115/1075671.1075742. https://aclanthology.info/pdf/H/H93/H93-1061.pdf.

[11] M. Baroni and A. Lenci, Distributional Memory: A General Framework for Corpus-Based Semantics, *Computational Linguistics* **36**(4) (2010), 673–721, ISSN 0891-2017. ISBN 10.1162/coli_a_00016. http://www.mitpressjournals.org/doi/pdfplus/10.1162/coli{_}a{_}00016.

[12] M. Baroni, G. Dinu and G. Kruszewski, Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors., in: *ACL*, 2014, pp. 238–247.

[13] O. Levy, Y. Goldberg and I. Dagan, Improving Distributional Similarity with Lessons Learned from Word Embeddings, *Transactions of the Association for Computational Linguistics* **3**(0) (2015), 211–225, ISSN 2307-387X. https://www.transacl.org/ojs/index.php/tacl/article/view/570.

[14] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, Distributed Representations of Words and Phrases and their Compositionality., in: *Advances in Neural Information Processing Systems*, Vol. cs.CL, 2013, pp. 3111–3119, ISSN 10495258. ISBN 2150-8097. doi:10.1162/jmlr.2003.3.4-5.951.

[15] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch and A. Joulin, Advances in Pre-Training Distributed Word Representations, in: *International Conference on Language Resources and Evaluation (LREC 2018)*, 2018. https://arxiv.org/pdf/1712.09405.pdfhttp://arxiv.org/abs/1712.09405.

[16] J. Pennington, R. Socher and C.D. Manning, Glove: Global vectors for word representation., in: *EMNLP*, Vol. 14, 2014, pp. 1532–1543.

[17] N. Shazeer, R. Doherty, C. Evans and C. Waterson, Swivel: Improving Embeddings by Noticing What's Missing, *arXiv preprint* (2016). https://arxiv.org/pdf/1602.02215.pdfhttp://arxiv.org/abs/1602.02215.

[18] I. Iacobacci, M.T. Pilehvar and R. Navigli, SENSEMBED: Learning Sense Embeddings for Word and Relational Similarity, in: *53rd Annual Meeting of the ACL*, 2015, pp. 95–105. ISBN 9781941643723. doi:10.3115/v1/P15-1010. http://anthology.aclweb.org/P/P15/P15-1010.pdf.

[19] X. Chen, Z. Liu and M. Sun, A Unified Model for Word Sense Representation and Disambiguation, in: *the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1025–1035. ISBN 978-1-937284-96-1. http://www.thunlp.org/site2/images/paper/D14-1110-chenxinxiong.pdf.

[20] S. Rothe and H. Schütze, AutoExtend: Extending Word Embeddings to Embeddings for Synsets and Lexemes, *ACL* (2015), 1793–1803. ISBN 9781941643723. https://arxiv.org/pdf/1507.01127.pdfhttp://arxiv.org/abs/1507.01127.

[21] M. Mancini, J. Camacho-Collados, I. Iacobacci and R. Navigli, Embedding Words and Senses Together via Joint Knowledge-Enhanced Training, *CoNLL* (2017). https://arxiv.org/pdf/1612.02703.pdfhttp://arxiv.org/abs/1612.02703.

[22] A. Bordes, N. Usunier, J. Weston and O. Yakhnenko, Translating Embeddings for Modeling Multi-Relational Data, *Advances in NIPS*, 2787–2795, ISSN 10495258. ISBN 9780874216561. doi:10.1007/s13398-014-0173-7.2.

[23] M. Nickel, L. Rosasco and T. Poggio, Holographic Embeddings of Knowledge Graphs, *AAAI* (2016), 1955–1961. ISBN 9781577357605. http://arxiv.org/abs/1510.04935.

[24] B. Shi and T. Weninger, ProjE: Embedding Projection for Knowledge Graph Completion, *eprint arXiv:1611.05425* (2016). http://arxiv.org/abs/1611.05425.

[25] P. Ristoski and H. Paulheim, RDF2Vec: RDF graph embeddings for data mining, in: *International Semantic Web Conference*, Vol. 9981 LNCS, 2016, pp. 498–514, ISSN 16113349. ISBN 9783319465227. https://ub-madoc.bib.uni-mannheim.de/41307/1/Ristoski{_}RDF2Vec.pdf.

[26] M. Schlichtkrull, T.N. Kipf, P. Bloem, R. van den Berg, I. Titov and M. Welling, Modeling Relational Data with Graph Convolutional Networks, 2018. https://pdfs.semanticscholar.org/11fa/3e8f58148abb1233376d2adac947c48e72c0.pdfhttp://arxiv.org/abs/1703.06103https://2018.eswc-conferences.org/wp-content/uploads/2018/02/ESWC2018{_}paper{_}4.pdf.

[27] L. Duong, H. Kanayama, T. Ma, S. Bird and T. Cohn, Learning Crosslingual Word Embeddings without Bilingual Corpora, in: *EMNLP-2016*, 2016, pp. 1285–1295. https://arxiv.org/pdf/1606.09403.pdfhttps://arxiv.org/abs/1606.09403.

[28] M. Ziemski, M. Junczys-Dowmunt and B. Pouliquen, The united nations parallel corpus v1. 0, in: *Language Resource and Evaluation*, 2016.

[29] R. Denaux and J.M. Gomez-Perez, Towards a Vecsigrafo: Portable Semantics in Knowledge-based Text Analytics, in: *International Workshop on Hybrid Statistical Semantic Understanding and Emerging Semantics @ISWC17*, 2017. https://pdfs.semanticscholar.org/b0d6/197940d8f1a5fa0d7474bd9a94bd9e44a0ee.pdf.

[30] L. Han, A. Kashyap, T. Finin, J. Mayfield and J. Weese, UMBC_EBIQUITY-CORE: Semantic Textual Similarity Systems, *Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics* **1** (2013), 44–52. ISBN 9781937284480. http://www.aclweb.org/anthology/S13-1005.

[31] M. Faruqui, Y. Tsvetkov, P. Rastogi and C. Dyer, Problems With Evaluation of Word Embeddings Using Word Similarity Tasks (2016). doi:10.18653/v1/W16-2506. https://arxiv.org/pdf/1605.02276.pdfhttp://arxiv.org/abs/1605.02276.

[32] T. Schnabel, I. Labutov, D. Mimno and T. Joachims, Evaluation methods for unsupervised word embeddings, in: *EMNLP*, Association for Computational Linguistics, 2015, pp. 298–307. http://anthology.aclweb.org/D/D15/D15-1036.pdf.

[33] H. Rubenstein and J.B. Goodenough, Contextual correlates of synonymy, *Communications of the ACM* **8**(10) (1965), 627–633, ISSN 00010782. doi:10.1145/365628.365657. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.893.7406{&}rep=rep1{&}type=pdfhttp://portal.acm.org/citation.cfm?doid=365628.365657.

[34] G.A. Miller and W.G. Charles, Contextual Correlates of Semantic Similarity, *Language and Cognitive Processes* **6**(1) (1991), 1–28, ISSN 14640732. ISBN 0169-0965. doi:10.1080/01690969108406936. http://www.tandfonline.com/doi/abs/10.1080/01690969108406936.

[35] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman and E. Ruppin, Placing search in context: the concept revisited, *ACM Transactions on Information Systems* **20**(1) (2002), 116–131, ISSN 10468188. ISBN 1581133480. doi:10.1145/503104.503110. http://www.cs.tau.ac.il/{~}ruppin/p116-finkelstein.pdfhttp://portal.acm.org/citation.cfm?doid=503104.503110.

[36] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Pas and A. Soroa, A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches, *Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL* (2009), 19–27, ISSN 1351-3249. ISBN 978-1-932432-41-1. doi:10.3115/1620754.1620758. http://www.aclweb.org/anthology/N09-1003.

[37] D. Yang and D.M.W. Powers, Verb Similarity on the Taxonomy of WordNet, *3rd International WordNet Conference* (2006), 121–128. http://dspace.flinders.edu.au/dspace/http://nlp.fi.muni.cz/gwc2006/proc/index.htmlhttp://semanticweb.kaist.ac.kr/conference/gwc/pdf2006/2..

[38] S. Baker, R. Reichart and A. Korhonen, An Unsupervised Model for Instance Level Subcategorization Acquisition, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)* (2014), 278–289. ISBN 9781937284961. http://www.aclweb.org/anthology/D14-1034http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.498.4645{&}rep=rep1{&}type=pdf.

[39] D. Gerz, I. Vulić, F. Hill, R. Reichart and A. Korhonen, SimVerb-3500: A Large-Scale Evaluation Set of Verb Similarity, *EMNLP* (2016). https://arxiv.org/pdf/1608.00869.pdfhttp://arxiv.org/abs/1608.00869.

[40] K. Radinsky, E. Agichtein, E. Gabrilovich and S. Markovitch, A word at a time, in: *Proceedings of the 20th international conference on World wide web - WWW '11*, ACM Press, New York, New York, USA, 2011, p. 337. ISBN 9781450306324. doi:10.1145/1963405.1963455. http://portal.acm.org/citation.cfm?doid=1963405.1963455.

[41] G. Halawi, G. Dror, E. Gabrilovich and Y. Koren, Large-scale learning of word relatedness with constraints, in: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12*, 2012, p. 1406, ISSN 9781450314626. ISBN 9781450314626. doi:10.1145/2339530.2339751. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.298.4921{&}rep=rep1{&}type=pdfhttp://dl.acm.org/citation.cfm?doid=2339530.2339751.

[42] E. Bruni, G. Boleda, M. Baroni and N.-K. Tran, Distributional semantics in technicolor, *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics* **1**(July) (2012), 136–145, ISSN 10504729.

ISBN 9781450310895. doi:10.1109/ICRA.2016.7487801. http://www.aclweb.org/anthology/P12-1015http://dl.acm.org/citation.cfm?id=2390544.

[43] F. Hill, R. Reichart and A. Korhonen, SimLex-999: Evaluating Semantic Models with (Genuine) Similarity Estimation, *Computational Linguistics* **41**(4) (2014), 665–695, ISSN 0891-2017. ISBN 9781608459858. http://www.mitpressjournals.org/doi/10.1162/COLI{_}a{_}00237http://arxiv.org/abs/1408.3456.

[44] M.-T. Luong, R. Socher and C.D. Manning, Better Word Representations with Recursive Neural Networks for Morphology, *CoNLL-2013* (2013), 104–113, ISSN 9781937284701. ISBN 9781937284701. http://www.aclweb.org/anthology/W13-3512.

[45] J. Camacho-Collados, M.T. Pilehvar, N. Collier and R. Navigli, SemEval-2017 Task 2: Multilingual and Cross-lingual Semantic Word Similarity, *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)* (2017), 15–26. http://www.aclweb.org/anthology/S17-2002.

[46] V. Liu and J.R. Curran, Web Text Corpus for Natural Language Processing, *Eacl* (2006), 233–240. ISBN 1932432590. https://aclanthology.coli.uni-saarland.de/pdf/E/E06/E06-1030.pdf.

[47] P. Koehn, Europarl : A Parallel Corpus for Statistical Machine Translation, *MT Summit* **11** (2005), 79–86, ISSN 9747431262. ISBN 9747431262. doi:10.3115/1626355.1626380. http://homepages.inf.ed.ac.uk/pkoehn/publications/europarl-mtsummit05.pdfhttp://mt-archive.info/MTS-2005-Koehn.pdf.

[48] J.M. Gómez-Pérez, R. Denaux, A. Garcia and R. Palma, A Holistic Approach to Scientific Reasoning Based on Hybrid Knowledge Representations and Research Objects, in: *Proceedings of Workshops and Tutorials of the 9th International Conference on Knowledge Capture (K-CAP2017), Austin, Texas, December 4th, 2017.*, 2017, pp. 47–49. http://ceur-ws.org/Vol-2065/paper09.pdf.

[49] A. Almuhareb and M. Poesio, Concept learning and categorization from the web, in: *Proceedings of the Cognitive Science Society*, Vol. 27, 2005. http://www.psych.unito.it/csc/cogsci05/frame/poster/1/f548-almuhareb.pdf.

[50] L. Aroyo and C. Welty, Truth is a lie: Crowd truth and the seven myths of human annotation, *AI Magazine* **36**(1) (2015), 15–24.

[51] J. Bian, B. Gao and T.-Y. Liu, Knowledge-Powered Deep Learning for Word Embedding. https://pdfs.semanticscholar.org/553a/6530b0802da9bec354d0a70fde254f6a5e36.pdf.

[52] R. Denaux, J. Biosca and J.M. Gomez-Perez, Framework for Supporting Multilingual Resource Development at Expert System, in: *Meta-Forum*, Lisbon, Portugal, 2016. http://www.meta-net.eu/events/meta-forum-2016/slides/31{_}denaux.pdf.

[53] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman and E. Ruppin, Placing search in context: The concept revisited, in: *Proceedings of the 10th international conference on World Wide Web*, ACM, 2001, pp. 406–414.

[54] J. Feng, M. Huang, Y. Yang and X. Zhu, GAKE: Graph Aware Knowledge Embedding, in: *COLING*, 2016, pp. 641–651. http://yangy.org/works/gake/gake-coling16.pdf.

[55] D. Gunning, V.K. Chaudhri, P.E. Clark, K. Barker, S.-Y. Chaw, M. Greaves, B. Grosof, A. Leung, D.D. McDonald, S. Mishra and Others, Project Halo UpdateâĂŤProgress Toward Digital Aristotle, *AI Magazine* **31**(3) (2010), 33–58.

[56] D. Kamholz, J. Pool and S.M. Colowick, PanLex: Building a Resource for Panlingual Lexical Translation, in: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, 2014, pp. 3145–3150. ISBN 978-2-9517408-8-4. https://panlex.org/pubs/etc/panlex-lrec2014.pdfhttp://www.lrec-conf.org/proceedings/lrec2014/pdf/1029{_}Paper.pdf.

[57] S. Lahiri, Complexity of Word Collocation Networks: A Preliminary Structural Analysis, in: *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, 2014, pp. 96–105. doi:10.3115/v1/E14-3011. https://arxiv.org/pdf/1310.5111.pdfhttp://aclweb.org/anthology/E14-3011.

[58] Omer Levy and Yoav Goldberg, Linguistic Regularities in Sparse and Explicit Word Representations, in: *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, Baltimore, Maryland, 2014, pp. 171–180. http://anthology.aclweb.org/W/W14/W14-16.pdf{#}page=181.

[59] Y. Lin, Z. Liu and M. Sun, Knowledge Representation Learning with Entities, Attributes and Relations, *IJCAI* (2016), 2866–2872. http://nlp.csai.tsinghua.edu.cn/{~}lyk/publications/ijcai2016{_}krear.pdf.

[60] T. Mikolov, Q.V. Le and I. Sutskever, Exploiting Similarities among Languages for Machine Translation, Technical Report, Google Inc., 2013, ISSN 10495258. ISBN 1309.4168. doi:10.1162/153244303322533223. https://arxiv.org/pdf/1309.4168.pdfhttp://arxiv.org/abs/1309.4168v1{%}5Cnhttp://arxiv.org/abs/1309.4168http://arxiv.org/abs/1309.4168.

[61] R. Navigli, Word Sense Disambiguation: A Survey, *ACM Comput. Surv* **41**(10) (2009). doi:10.1145/1459352.1459355. http://promethee.philo.ulg.ac.be/engdep1/download/bacIII/ACM{_}Survey{_}2009{_}Navigli.pdf.

[62] D. Smilkov, G. Brain, N. Thorat, C. Nicholson, E. Reif, F.B. Viégas and M. Wattenberg, Embedding Projector: Interactive Visualization and Interpretation of Embeddings, in: *Workshop on Interpretable Machine Learning in Complex Systems at NIPS*, 2016. https://arxiv.org/pdf/1611.05469.pdf.

[63] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury and M. Gamon, Representing Text for Joint Embedding of Text and Knowledge Bases (2015), 1499–1509. https://www.aclweb.org/anthology/D/D15/D15-1174.pdf.

[64] Z. Wang, J. Zhang, J. Feng and Z. Chen, Knowledge Graph and Text Jointly Embedding, *EMNLP* **14** (2014), 1591–1601. ISBN 9781937284961. https://pdfs.semanticscholar.org/f108/973a380bddeed5cd4eac95670194db667441.pdf.

[65] J. Weston, A. Bordes, O. Yakhnenko and N. Usunier, Connecting Language and Knowledge Bases with Embedding Models for Relation Extraction (2013), 1366–1371. https://pdfs.semanticscholar.org/4f5c/d4c2d81db5c52f952589a8d52bba16962707.pdf.