Inducing Conceptual Embedding Spaces from Wikipedia

Gerard de Melo
Rutgers University
New Brunswick, NJ
USA
gdm@demelo.org

ABSTRACT

The word2vec word vector representations are one of the most well-known new semantic resources to appear in recent years. While large sets of pre-trained vectors are available, these focus on frequent words and multi-word expressions but lack sufficient coverage of named entities. Moreover, Google only released pre-trained vectors for English. In this paper, we explore an automatic expansion of Google's pre-trained vectors using Wikipedia, adding millions of concepts and named entities in over 270 languages. Our method enables all of these to reside in the same vector space, thus flexibly facilitating cross-lingual semantic applications.

Keywords

conceptual knowledge; semantic representations; Wikipedia

1. INTRODUCTION

Motivation. Over the course of the last decade, we have witnessed the growing popularity of large-scale semantic databases and knowledge graphs, which are now important assets for many big companies. Recently, however, another form of semantic resource has been attracting enormous attention. The word2vec models [32], in conjunction with the corresponding tools and data released by Google Inc., allow us to map words to dense vector representations such that similar words are mapped to similar vectors in the corresponding vector space. This approach has received significant attention not only in natural language processing but also in the broader community of practitioners working with textual data and semantics. The three word2vec-related papers published by Mikolov et al. in 2013 have already garnered more than 3,000 citations according to Google Scholar.

Such representations are also referred to as neural embeddings, since neural networks are used to learn a mapping from objects into a vector space. The resulting vectors exhibit a number of appealing properties. Not only do their cosine similarities correlate well with human judgments of word similarity and relatedness. They have also been found to reflect the commonsense knowledge and semantic knowledge necessary for analogical reasoning [43]. Still, knowledge graphs such as YAGO [24] and MENTA [14] have important advantages with regard to their coverage of entities and their support for multilingual and cross-lingual lookups.

Overview and Contributions. In this work, we investigate the intersection of these two directions of semantic resources, creating vector representations not just for words but also for the millions of multilingual named entities and concepts described by Wikipedia. Specifically, we exploit the fact that the pre-trained vectors released by Google already contain millions of frequent words and expressions covered by the English Wikipedia as well as the fact that the English Wikipedia contains numerous other named entities and is linked to the over 200 other language editions of Wikipedia. To facilitate this expansion process, we rely on MENTA [14], a large multilingual knowledge graph that conveniently transforms these over 200 language editions of Wikipedia, in conjunction with the English WordNet [18], into a single unified hierarchically organized network of entities and natural language terms.

Drawing on word2vec as well as MENTA, we obtain a new semantic embedding resource covering millions of words and names in over 200 languages. For each of these, it provides a numerical vector that can be used to quickly compute the similarity between two nouns or entity names, even if they are given in completely different languages. Experiments on a series of semantic relatedness tasks show that this approach of drawing on knowledge bases compares favourably against state-of-the-art approaches.

2. BACKGROUND

Semantic Representations. Traditional knowledge representation has been inherently symbolic, making use of discrete symbols such as entity IDs or URIs to represent entities and their relationships. In many settings, such representations seem ideal to reliably capture and store a given set of facts. Using symbolic subject-predicate-object triples, we can easily represent that *Kobe* is located in *Japan* [8].

In many settings, however, especially when relying on machine learning and certain kinds of artificial intelligence methods, it can be important to exploit less explicit signals about entities. For instance, based on word frequency counts, we may have an indication that there is a salient association between *Kobe* and *beef*, but we may lack more detailed knowledge about the nature of this association.

Machine learning algorithms also have trouble coping with symbols that never or only rarely appeared in the training

WWW 2017 Companion, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4914-7/17/04.
http://dx.doi.org/10.1145/3041021.3054144

data. If every symbol is treated as completely distinct, then machine learning techniques will often fail to make sensible decisions for any new symbols appearing at test time. This is an important challenge when working with textual data, where one often faces the more specific problem of out-of-vocabulary words, e.g. when machine translation systems encounter new, previously unobserved words, and have no clue about how to translate them.

Word Embeddings. In the past few years, word embeddings have quickly become the most well-known computational resource for representing word meanings [44, 32]. These word embeddings embed words in an n-dimensional Euclidean vector space. While the specific dimensions do not normally carry any particular meaning, the word vector mappings are optimized in a way that words that are similar in meaning are mapped to similar vectors. Based on this property, machine learning methods are equipped with semantic information that enables them to transfer information about observed words to previously unobserved ones. This has proven particularly valuable for deep learning and other neural models, which tend to favor dense representations over the rather sparse one-hot bag-of-words vector representations. The latter have been prominent both in information retrieval and when transforming text into feature vector representations suitable support vector machines [11].

Word vectors have quickly surpassed most previously used linguistic resources in popularity. While part of this may be hype, empirical studies show that neural network-based prediction models for creating word vectors indeed outperform several previous efforts at distributional semantics [1], although it has also been shown that with the lessons we have learned from these newer models, traditional matrix methods can also be adapted to obtain similar results [27].

Another factor that has contributed to the widespread use of word2vec is the simplicity of working with word vectors. For one, vector representations of words are elegantly simple to use and understand, requiring only basic vector arithmetic to obtain state-of-the-art results on certain lexical tasks. At the same time, Google's decision to release word vectors pre-trained on a very large Google News dataset has contributed to facilitating their widespread adoption. Interested practitioners can simply download the pre-trained word vectors and immediately start mapping words to vectors without having to procure any large text corpus for training.

The original word2vec methods optimize the vectors by relying on contextual information in a very large text corpus [32]. The pre-trained word2vec vectors[1] cover 3,000,000 terms, out of which 929,022 are single token terms and 2,070,978 are multi-word units. The latter were added to the lexicon using a frequency heuristic, and some are named entities, while others are short expressions and phrases. Still, these pre-trained embeddings lack many less frequent named entities that may not appear often enough in regular text corpora. In particular, word2vec is normally run with a frequency cut-off threshold, dropping all words that are less frequent than this threshold. Although this threshold is a parameter that can be modified, word2vec needs to observe a word in a significant number of training contexts in order to be able to derive appropriate vector representations for it.

Moreover, Google has only released vectors for English, and training vectors for other languages requires customization of the code to deal with issues such as tokenization and non-Latin characters. If one does make these modifications, one still arrives at vectors that are incompatible with the existing English vectors. Our vectors in contrast, allow for cross-lingual comparisons. We will thus be able to compare a German word such as *Tragfläche* with an English expression such as *Lift-to-drag ratio*.

[1]Available from Google at `https://code.google.com/p/word2vec/`.

3. ALGORITHMIC APPROACH

3.1 Concepts and Entities

Our work attempts to exploit WIkipedia (via the MENTA resource [14]) to provide a simple yet effective means of quickly obtaining high-quality vectors for millions of entities in over 200 languages.

Our approach assumes a universe $\mathcal{U}$ of objects that we wish to map to vector representations in an n-dimensional Euclidean space $[0,1]^n \subseteq \mathbb{R}^n$. The objects to be mapped can be linguistic expressions as well as sense or entity or concept identifiers. Linguistic expressions are strings coupled with a language identifier. Thus, the English word *coin* is treated as distinct from the French word *coin* (corner). In addition to such linguistic identifiers, which can be ambiguous, we separately consider entity or concept identifiers that are intended to be unambiguous. Thus, there is not just an entry for the English term *Georgia*, which is ambiguous (possibly referring to a country or to a US state, among other meanings), but also specific identifiers referring specifically to the country and state, as identified by their respective Wikipedia article titles.

We further assume an input set $X \subseteq \mathcal{U}$, and a set X_0 of objects $x \in X_0$ for which we have prior knowledge in the form of coherent high-quality vector embeddings $\widetilde{\mathbf{v}}_x \in [0,1]^n$. In our experiments, we will use Google's original word2vec embeddings, trained on a large Google News dataset, for this purpose.

3.2 Training Objective

Our goal is to produce embeddings $\mathbf{v}_i$ for all $x_i \in X$. In particular, for these vectors to be useful in downstream machine learning algorithms and semantic tasks, we seek to achieve this subject to the condition that dot products $\mathbf{v}_i^t \mathbf{v}_j$ for vectors $\mathbf{v}_i$, $\mathbf{v}_j$ of object pairs x_i, x_j reflect some notion of similarity between x_i and x_j.

As evidence of object similarities, we take as input a set of known object relationships $R \subseteq X \times X$, with corresponding weights $w_R(x_i, x_j) \in \mathbb{R}$. These object relationships can come from one or more knowledge bases.

We first generalize $w_R(x_i, x_j)$ to a more general function

$$w(x_i, x_j) = \begin{cases} w_R(x_i, x_j) & (i,j) \in R \\ -1 & \text{otherwise} \end{cases} \quad (1)$$

Our training objective is then to maximize the following function:

$$\sum_{i=1}^{|X|} \sum_{j=1}^{|X|} w(x_i, x_j)\, \mathbf{v}_i^T \mathbf{v}_j \tag{2}$$

3.3 Optimization

For optimization, we rely on stochastic gradient ascent steps [20], in which we iterate over R, in randomized order, while also using negative sampling [32] to draw examples with negative weights. Thus, the vectors of random positive pairs from R are repeatedly adapted so as to increase their similarity, while random negative pairs are encouraged to have a lower similarity.

Before starting the optimization procedure, we preinitialize the vectors $\mathbf{v}_i$ for all x_i covered by the initial prior knowledge from Google's pre-released word vectors, while all other vectors are initially set to zero, so as to reduce the effect that these vectors have on other vectors during the initial warming up phase. In each iteration, we select a random triple from our knowledge base as input. This is a positive example. For negative sampling, we also randomly replace the subject or object with a randomly chosen entity and treat this as a negative sample. We compute the gradient of our objective function for these two samples and make a small gradient optimization step to update the involved vectors. The size of these steps is determined by the gradient itself and by the learning rate, which, as is common practice, decreases over time [20], as explained in our experiments.

4. DATA AND TRAINING

Vectors. Our input word vectors are the well-known word2vec Skip-Gram with Negative Sampling ones trained on a large Google News corpus consisting of around 100 billion word tokens, and released for public use by Google[2].

Knowledge Base. For the relationships between objects, we rely on MENTA [14], which aggregates and transforms information from WordNet [18] and over 270 language editions of Wikipedia. While Wikipedia already contains rich semantic information, MENTA ensures that this knowledge is connected to disambiguated lexicon entries in WordNet, which is helpful for our method, as it enables us to reduce the affect of noise due to ambiguous words and entries. For instance, the original input vectors only contain a single entry for an ambiguous name such as *Georgia*, while our resource needs to create separate vectors for the US State of Georgia vs. the country Georgia. We also include the original WordNet as input in order to have the upper-level hierarchy and lexicon that MENTA builds upon.

An overview of the input relationships is given in Table 1. The `lexicalization` relation connects entities to their linguistic expressions, e.g. connecting the entity `United States of America` to a number of English linguistic expressions, including *United States of America*, *United States*, and *USA*, as well as to numerous non-English linguistic expressions. The `instance of` relation connects an entity such as `New York City` to a class such as `city`, while `subclass of` and the similar notion of WordNet hypernymy connect `city` to the more general, i.e., less specific class `municipality`.

For the relationship weights $w(x_i, x_j)$, we simply use the original weights given by MENTA, or 1 for unweighted relationships from knowledge sources such as WordNet that do not provide weights.

Table 1: Input relationships.

Relationship	Count (Word Pairs)
Lexicalization	21,257,844
Instance of	5,519,065
Subclass of	540,952
WordNet Hypernymy	94,769

Training. During training, we use stochastic gradient ascent with a starting learning rate of 0.1, which decreases by a factor of 12 in every epoch over the knowledge base inputs. In our experiments, we ran our algorithm for just 5 epochs, as the results appear to stabilize rather quickly due to the preinitialization.

5. EMPIRICAL EVALUATION

We next report the results of said training process and evaluate the output using a number of evaluation datasets.

5.1 Coverage

In Table 2, we provide an overview of the resulting word vectors in terms of their coverage. In total, our data covers 45 languages with at least 50,000 entries, 185 languages with at least 1000 entries, and 297 languages with at least 100 words.

In addition to the named entities and words in different languages, our approach also yields vectors for 5,722,950 disambiguated Wikipedia concepts (MENTA identifiers each connected to Wikipedia pages in one or more languages) and 114,538 disambiguated WordNet synsets.

In Table 3, we see that our output vectors show substantial improvements in coverage on the Stanford Rare Words word relatedness dataset [31], which also entail improvements in the Spearman correlation with human judgements. We rely on the standard Spearman ρ correlation coefficient, computed with proper tie-breaking.

5.2 Semantic Relatedness

Apart from the Stanford Rare Words dataset mentioned above, we further evaluated our results on the French JI-65 word relatedness dataset in Table 4, showing improvements over previous work.

Next, in Table 5 we evaluated cross-lingual relatedness results on the multilingual version of the RG65 data produced by Camacho-Collados et al. [3]. Since our resource only covers nominal concepts covered by Wikipedia, our resource lacks certain terms such as the German *grinsen* (to grin). Still, the results show that we obtain a comparable vector quality, while covering a larger vocabulary.

5.3 Verbal IQ Synonymy Evaluation

Next, we considered the IQ test questions from Wang et al. 2015 [45]. In their paper, they consider several different types of questions, each modeled separately. While

[2] `https://code.google.com/p/word2vec/`

Table 2: Coverage comparison, with total number of terms and multi-word terms (those containing a space character)

Vectors	Language	# total terms	# MWEs
Original Google News word2vec	English	3,000,000	2,070,978
Our Approach	English	6,477,502	5,100,963
	French	1,117,508	866,470
	German	880,206	583,267
	Spanish	762,552	574,718
	Portuguese	621,062	441,510
	Italian	620,592	432,048
	Polish	561,223	380,170
	Russian	542,213	365,373
	Dutch	509,650	279,227
	Japanese	486,883	21,757
	Swedish	357,827	237,554
	...	...	...
	All	18,704,387	12,395,795

Table 3: Spearman's Correlation and Coverage on the Stanford Rare Word dataset by Luong et al. 2013 [31].

Vectors	Spearman ρ	Pair Coverage
Original word2vec	0.421	1863 (91.6%)
Our Approach	0.450	1987 (97.7%)

Table 4: Spearman's Correlation on the French JI-65 dataset, a translation of the English RG65 dataset.

Vectors	Spearman ρ
Granada et al. 2014 [21]	0.52
Faruqui & Dyer 2014 [17]	0.61
Camacho-Collados et al. 2015 [3]	0.71
Our Approach	0.724

Table 5: Spearman correlations on cross-lingual RG65 data by Camacho-Collados et al. 2015 [3].

Language 1	Language 2	Vectors	Spearman ρ
English	French	CL-MSR 2.0	0.30
		PMI-SVD pivot	0.76
		word2vec pivot	0.75
		ADW pivot	0.80
		Camacho-Collados et al. 2015 [3] pivot	0.83
		Camacho-Collados et al. 2015 [3]	0.83
		Our Approach	0.815
English	German	PMI-SVD pivot	0.72
		word2vec pivot	0.69
		ADW pivot	0.73
		Camacho-Collados et al. 2015 [3] pivot	0.73
		Camacho-Collados et al. 2015 [3]	0.76
		Our Approach	0.764
French	German	PMI-SVD pivot	0.65
		word2vec pivot	0.77
		ADW pivot	0.72
		Camacho-Collados et al. 2015 [3] pivot	0.79
		Camacho-Collados et al. 2015 [3]	0.83
		Our Approach	0.782

our objective does not include a model of relations such as antonymy, we can easily evaluate our vectors on their synonymy dataset. For this, we simply evaluate the cosine similarities of words' vector representations to choose the most similar answer.

The results in Table 6 show that our approach not only improves over the original results, which serve as our input. Surprisingly, both of these outperform the method by Wang et al. [45], the creators of the dataset, although their method is supervised in the sense that their training objective explicitly attempts to reproduce large amounts of semantic relationships collected from dictionaries. We additionally outperform the Multi-Sense Vectors from Huang et al. 2012 [25], in terms of the evaluation results reported by Wang et al. 2015 [45]. Interestingly, in their study, humans underperformed on this task, although we conjecture that this might be due to the crowdsourcing approach chosen for assessing human performance.

5.4 Reader's Digest Word Choice Problems

Next, we experimented with a German-language word choice quiz dataset[3]. This dataset contains 984 problem instances collected from the 2001 to 2005 editions of the German version of Reader's Digest Magazine, where they appear as "Word Power" problems.

Given a target word and four candidate phrases, the goal is to select the phrase that describes the target word. Consider the following English language examples.

gourmet	**dale**
a) enjoys cooking	a) plain
b) has indigestion	b) retreat
c) has an expert appreciation of food	c) shelter
d) is hungry	d) valley

Here, the correct answers are c) for *gourmet* and d) for *dale*. Picking the right answers hinges not only on our familiarity with a given word and its meaning but also on our ability to relate it to the descriptions provided as candidate answers.

We rely on the simple method of computing cosine similarities between the target word and the candidate answers. Some answers are individual words or expressions already covered in our data, in which case this is simple. If a candidate answer, however, consists of multiple words that are not covered in our data as a multi-word expression, we simply use the maximum cosine similarity between any of the words in the answer phrase and the target word.

We assess the accuracy as the sum of scores over all problem instances divided by the number of problem instances. Following the convention from previous work [33], these evaluation scores are 1 if the correct answer is ranked highest among all candidates by our method, 0 if it is not ranked highest, and $\frac{1}{n}$ if our method's top ranked answers form a tie of n answers with the same similarity score.

The results are provided in Table 7. We see reasonable results, outperforming the German-English vectors from Chandar et al. 2014 [4]. Note that by random guessing, as for the original word2vec vectors, which do not cover German, one obtains only 25%. Of course, our results are not perfect, since we are applying simple word vectors to a task that requires understanding entire linguistic phrases. Better results could easily be obtained by improving the linguistic analysis of candidate answers, for instance by performing lemmatization, stop word removal or interpretation, and compound splitting, which, of course, is particularly helpful for German with its notoriously long compound nouns. After that, one could then use our vectors to obtain more reliable similarity scores.

[3] `https://www.ukp.tu-darmstadt.de/data/semantic-relatedness/german-word-choice-problems/`

6. RELATED WORK

Knowledge Bases. Many well-known semantic resources are symbolic in the sense that there are discrete items with discrete properties. While traditional knowledge bases often used knowledge representation formalisms closely related to formal logics [12, 41], modern knowledge graphs tend to rely on simpler labeled graphs, in which nodes represent arbitrary entities and edges represent their connections. Examples of such knowledge graphs include Freebase and YAGO [24]. Beyond simple subject-predicate-object triples, such knowledge graphs can also incorporate multimodal data [42]. Many current knowledge graphs further include linguistic and lexical knowledge [38, 9], and thus can be connected to resources such as WordNet [18], while even non-linguistic ones are also based on Wikipedia. Due to these connections, it has been possible to create massively multilingual knowledge graphs that cover entities and concepts in numerous languages [13, 14].

Embeddings from Wikipedia. Different approaches have been presented to go beyond discrete symbolic knowledge. Distributional semantic approaches exploit cooccurrence patterns between words to induce high-dimensional vector spaces [39]. Techniques such as latent semantic analysis [15] rely on document-word matrices. Gutiérrez et al. 2016 [22] proposed a method to create and use multilingual topic models, which encode probabilistic distributions over concepts.

In recent years, low-dimensional vector embeddings of words have proven very popular as representations [44]. When releasing word2vec [32] to the public, Google itself published not only word vectors, but also vectors for Freebase entity identifiers. These were obtained by detecting and disambiguating named entities using an automatic entity linking approach and then applying the standard word2vec SGNS model. However, this approach only covers named entities frequent enough to appear in the corpus and the resource that they produced does not share the same embedding space with the regular word vectors that they released. Additionally, it uses custom entity IDs that are no longer generally supported, given that Freebase has been retired and has not been accepting updates for a long while now. A related tool has been released (`https://github.com/idio/wiki2vec`) to enable using Wikipedia as a corpus for DBpedia vectors. However, this approach only addresses a single language at a time and cannot support cross-lingual similarity computations.

Relational Learning. Another related but distinct line of work has focused on relational learning and link prediction. The goal of this is to learn a model that is able to predict relationships between entities. In particular, this can involve predicting whether a given relationship holds between two entities, or, e.g., given subject `France` and relation `hasCapitalCity`, the goal could be to predict the entity `Paris`. In

Table 6: Accuracy results on MSR Synonym IQ test questions

Dataset	Accuracy
Avg. Human (as reported by Wang et al. 2015 [45])	50.38%
Multi-Sense Model	50.00%
Supervised Model by Wang et al. 2015 [45]	60.78%
Original word2vec	62.75%
Our Approach	66.67%

Table 7: Accuracy results on German word choice problems

Vectors	Accuracy
Random guessing / Original word2vec input vectors	25.00%
Chandar et al. 2014 [4] En-De Vectors	27.35%
Our Approach	49.70%

recent years, the methods that accomplish such tasks have often relied on tensor modeling as in RESCAL [34] and the Neural Tensor Network model [40], or on even simpler vector and matrix representations. The TransE model [2], for instance, treats relations as simple translations between two vectors. More recently, numerous variations of TransE have been proposed to account for specific phenomena. TransH [46], for instance, models relations as translations on hyperplanes. TransR [29] adds extra projections of entity vectors for each specific relation, or, in the CTransR variant, for each cluster of relations. PTransE [28] attempts to consider inference via property paths to improve the prediction of a triple (for example, x `bornInCity` y, y `cityInState` z helps us predict x `bornInState` z). Although some of these methods incorporate learning vector representations of entities, these are just a byproduct. It has been shown that link prediction methods lead to substantially different vector representations than those produced by corpus-based word vector learning methods [6]. In particular, such representations tend not to yield high correlations with human similarity judgments [6].

Extending Embeddings. There have been several previous efforts to extend pre-existing word vectors using additional resources. Some approaches rely on linguistic resources. For instance, Rothe & Schütze [37] presented an autoencoder neural network that can be used to derive word vectors for WordNet lexemes and synsets from undisambiguated word vectors. Faruqui et al. 2015 [16] showed that lexical resources such as WordNet, PPDB, and FrameNet can be used to improve the quality of word vectors. Previous work has also shown that translation information such as from Wiktionary can be used to extend word2vec and GloVe word vectors [10].

Another line of work follows a more generic data-driven approach. In particular, Loza Mencía et al. showed that jointly representing words, documents, and document labels can lead to improved embeddings [30]. Chen & de Melo 2015 [5] presented a joint optimization framework that simultaneously learns word embeddings from a corpus and from extractions mined from text, focusing especially on lists of similar words. Chen et al. 2016 [6] extended this idea by jointly modeling fact extractions using relation-specific matrices. Their approach jointly optimizes for link prediction of the sort mentioned above and corpus-based word vector learning.

However, none of these approaches come even close in terms of coverage to the large-scale extension by many millions of named entities that we present here.

Entity and Fact Similarity. There have also been other approaches to computing similarities between Wikipedia entries. Paulheim 2013 [35] developed the DBpedia FindRelated service to find related entities. It relies on SVMs trained using external information coming from Web search engines and thus benefits from additional data that we do not consider in our approach. Pereira Nunes et al. 2012 [36] relied on a graph relation-based approach. Unlike these approaches, our method jointly creates representations both of Wikipedia entries and of the terms that they represent and can thus be evaluated using word similarity datasets, while for DBpedia entries, to the best of our knowledge, no gold standard relatedness dataset exists, although the creation of such datasets has been considered before as future work [35]. Another advantage of our approach is that it is also easily possible to obtain assessments for ambiguous terms such as *Java* or *Jaguar*, and our method directly supports words and named entities given in hundreds of languages.

Finally, a related but distinct problem is that of grouping together semantically similar facts [19, 7, 23, 26]. Here the objects to be related are entire subject-predicate-object statements rather than just entities. Methods that achieve this are useful, for instance, in user interfaces for browsing structured data.

7. CONCLUSION

We have shown that by drawing on a joint optimization objective, we are able to leverage Wikipedia and its transformation in the MENTA knowledge graph to induce a large joint embedding space. This leads to a novel kind of resource that can not only replace the original word vectors but also supports certain kinds of lexical semantic operations that are not as easily achievable with the original Wikipedia or MENTA resources. Importantly, we can compute similarities between arbitrary entities, being able to use either disambiguated identifiers or ambiguous words and names in different languages. Overall, we obtain a resource that is compatible with the original word2vec representation space but adds millions of named entities in over 200 languages.

8. REFERENCES

[1] M. Baroni, G. Dinu, and G. Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[2] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26*, pages 2787–2795. 2013.

[3] J. Camacho-Collados, M. T. Pilehvar, and R. Navigli. A unified multilingual semantic representation of concepts. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 741–751. The Association for Computer Linguistics, 2015.

[4] S. Chandar A P, S. Lauly, H. Larochelle, M. Khapra, B. Ravindran, V. C. Raykar, and A. Saha. An autoencoder approach to learning bilingual word representations. In *Proceedings of NIPS 2014*, pages 1853–1861. Curran Associates, Inc., 2014.

[5] J. Chen and G. de Melo. Semantic information extraction for improved word embeddings. In *Proceedings of the NAACL Workshop on Vector Space Modeling for NLP*, 2015.

[6] J. Chen, N. Tandon, C. D. Hariman, and G. de Melo. Webbrain: Joint neural learning of large-scale commonsense knowledge. In *Proceedings of ISWC 2016*, 2016.

[7] G. Cheng, T. Tran, and Y. Qu. Relin: Relatedness and informativeness-based centrality for entity summarization. In L. Aroyo, C. Welty, H. Alani, J. Taylor, A. Bernstein, L. Kagal, N. F. Noy, and E. Blomqvist, editors, *International Semantic Web Conference*, volume 7031 of *Lecture Notes in Computer Science*, pages 114–129. Springer, 2011.

[8] G. de Melo. Not quite the same: Identity constraints for the Web of Linked Data. In M. desJardins and M. L. Littman, editors, *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI 2013)*, pages 1092–1098, Menlo Park, CA, USA, 2013. AAAI Press.

[9] G. de Melo. Lexvo.org: Language-related information for the Linguistic Linked Data cloud. *Semantic Web*, 6(4):393–400, August 2015.

[10] G. de Melo. Wiktionary-based word embeddings. In *Proceedings of MT Summit XV*, 2015.

[11] G. de Melo and S. Siersdorfer. Multilingual text classification using ontologies. In G. Amati, C. Carpineto, and G. Romano, editors, *Advances in Information Retrieval, 29th European Conference on IR Research, ECIR 2007, Rome, Italy, April 2-5, 2007, Proceedings*, volume 4425 of *Lecture Notes in Computer Science*, pages 541–548. Springer, 2007.

[12] G. de Melo, F. Suchanek, and A. Pease. Integrating YAGO into the Suggested Upper Merged Ontology. In *Proceedings of the 20th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2008)*. IEEE Computer Society, Los Alamitos, CA, USA, 2008.

[13] G. de Melo and G. Weikum. Towards a universal wordnet by learning from combined evidence. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM 2009)*, pages 513–522, New York, NY, USA, 2009. ACM.

[14] G. de Melo and G. Weikum. Taxonomic data integration from multilingual Wikipedia editions. *Knowledge and Information Systems*, 39(1):1–39, April 2014.

[15] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407, 1990.

[16] M. Faruqui, J. Dodge, S. K. Jauhar, C. Dyer, E. Hovy, and N. A. Smith. Retrofitting word vectors to semantic lexicons. In *Proceedings of NAACL 2015*, 2015.

[17] M. Faruqui and C. Dyer. Improving vector space word representations using multilingual correlation. In *Proceedings of EACL*, 2014.

[18] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.

[19] T. Franz, A. Schultz, S. Sizov, and S. Staab. *The Semantic Web - ISWC 2009: 8th International Semantic Web Conference, ISWC 2009, Chantilly, VA, USA, October 25-29, 2009. Proceedings*, chapter TripleRank: Ranking Semantic Web Data by Tensor Decomposition, pages 213–228. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[20] I. Goodfellow, Y. Bengio, and A. Courville. Deep learning. Book in preparation for MIT Press, 2016.

[21] R. Granada, C. T. dos Santos, and R. Vieira. Comparing semantic relatedness between word pairs in portuguese using wikipedia. In J. Baptista, N. J. Mamede, S. Candeias, I. Paraboni, T. A. S. Pardo, and M. das Graças Volpe Nunes, editors, *Computational Processing of the Portuguese Language - 11th International Conference, PROPOR 2014, São Carlos/SP, Brazil, October 6-8, 2014. Proceedings*, volume 8775 of *Lecture Notes in Computer Science*, pages 170–175. Springer, 2014.

[22] E. D. Gutiérrez, E. Shutova, P. Lichtenstein, G. de Melo, and L. Gilardi. Detecting cross-cultural differences using a multilingual topic model. *Transactions of the Association for Computational Linguistics (TACL)*, 4:47–60, 2016.

[23] J. Hees, T. Roth-Berghofer, R. Biedert, B. Adrian, and A. Dengel. *Search Computing: Broadening Web Search*, chapter BetterRelations: Collecting Association Strengths for Linked Data Triples with a Game, pages 223–239. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[24] J. Hoffart, F. M. Suchanek, K. Berberich, E. Lewis-Kelham, G. de Melo, and G. Weikum.

YAGO2: Exploring and querying world knowledge in time, space, context, and many languages. In S. Srinivasan, K. Ramamritham, A. Kumar, M. P. Ravindra, E. Bertino, and R. Kumar, editors, *Proceedings of the 20th International World Wide Web Conference (WWW 2011)*, pages 229–232, New York, NY, USA, 2011. ACM.

[25] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng. Improving Word Representations via Global Context and Multiple Word Prototypes. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2012.

[26] J. Huelss and H. Paulheim. What sparql query logs tell and do not tell about semantic relatedness in lod – or: The unsuccessful attempt to improve the browsing experience of dbpedia by exploiting query logs. In F. Gandon, C. GuÃl'ret, S. Villata, J. G. Breslin, C. Faron-Zucker, and A. Zimmermann, editors, *ESWC (Satellite Events)*, volume 9341 of *Lecture Notes in Computer Science*, pages 297–308. Springer, 2015.

[27] O. Levy, Y. Goldberg, and I. Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015.

[28] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, and S. Liu. Modeling relation paths for representation learning of knowledge bases. In *Proceedings of EMNLP*, 2015.

[29] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings AAAI 2015*. AAAI Press, 2015.

[30] E. Loza Mencía, G. de Melo, and J. Nam. Medical concept embeddings via labeled background corpora. In *Proceedings of the 10th Language Resources and Evaluation Conference (LREC 2016)*, Paris, France, 2016.

[31] M.-T. Luong, R. Socher, and C. D. Manning. Better word representations with recursive neural networks for morphology. In *CoNLL*.

[32] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[33] S. Mohammad, I. Gurevych, G. Hirst, and T. Zesch. Cross-lingual distributional profiles of concepts for measuring semantic distance. In *2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, June 2007.

[34] M. Nickel, V. Tresp, and H.-P. Kriegel. A three-way model for collective learning on multi-relational data. In L. Getoor and T. Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 809–816, New York, NY, USA, June 2011. ACM.

[35] H. Paulheim. Dbpedianyd a silver standard benchmark dataset for semantic relatedness in dbpedia. In *The 12th International Semantic Web Conference (ISWC2013)*, 2013.

[36] B. Pereira Nunes, R. Kawase, S. Dietze, D. Taibi, M. Antonio, and W. Nejdl. Can entities be friends. In *Proceedings of WOLE, in conjunction with ISWC 2012, volume 906 of CEURWS.org*, pages 45–57, 2012.

[37] S. Rothe and H. Schütze. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1793–1803, Beijing, China, July 2015. Association for Computational Linguistics.

[38] J. Rouces, G. de Melo, and K. Hose. FrameBase: Representing n-ary relations using semantic frames. In *Proceedings of ESWC 2015*, pages 505–521, 2015.

[39] H. Schütze. Word space. In *Advances in Neural Information Processing Systems 5 (NIPS 1992)*, 1993.

[40] R. Socher, D. Chen, C. D. Manning, and A. Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems 26*, pages 926–934. 2013.

[41] G. Sutcliffe, M. Suda, A. Teyssandier, N. Dellis, and G. de Melo. Progress towards effective automated reasoning with world knowledge. In H. W. Guesgen and R. C. Murray, editors, *Proceedings of the 23rd International FLAIRS Conference*, pages 110–115, Menlo Park, CA, USA, 2010. AAAI Press.

[42] N. Tandon, G. de Melo, A. De, and G. Weikum. Knowlywood: Mining activity knowledge from Hollywood narratives. In *Proceedings of CIKM 2015*, 2015.

[43] G. Z. Tomas Mikolov, Scott Wen-tau Yih. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-2013)*. Association for Computational Linguistics, May 2013.

[44] J. Turian, L. Ratinov, and Y. Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of ACL 2010*, pages 384–394, 2010.

[45] H. Wang, B. Gao, J. Bian, F. Tian, and T. Liu. Solving verbal comprehension questions in IQ test by knowledge-powered word embedding. *CoRR*, abs/1505.07909, 2015.

[46] Z. Wang, J. Zhang, J. Feng, and Z. Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of AAAI 2014*, pages 1112–1119, 2014.