Extending WordNet with Fine-Grained Collocational Information via Supervised Distributional Learning

Luis Espinosa-Anke[1], Jose Camacho-Collados[2], Sara Rodríguez-Fernández[1], Horacio Saggion[1], Leo Wanner[3,1]

[1]NLP Group, Universitat Pompeu Fabra
firstname.lastname@upf.edu
[2]Department of Computer Science, Sapienza University of Rome
collados@di.uniroma1.it
[3]Catalan Institute for Research and Advanced Studies (ICREA)

Abstract

WordNet is probably the best known lexical resource in Natural Language Processing. While it is widely regarded as a high quality repository of concepts and semantic relations, updating and extending it manually is costly. One important type of relation which could potentially add enormous value to WordNet is the inclusion of collocational information, which is paramount in tasks such as Machine Translation, Natural Language Generation and Second Language Learning. In this paper, we present ColWordNet (CWN), an extended WordNet version with fine-grained collocational information, automatically introduced thanks to a method exploiting linear relations between analogous sense-level embeddings spaces. We perform both intrinsic and extrinsic evaluations, and release CWN for the use and scrutiny of the community.

1 Introduction

The embedding of cues about how we perceive concepts and how these concepts relate and generalize across different domains gives knowledge resources the capacity of generalization, which lies at the core of human cognition (Yu et al., 2015) and is also central to many Natural Language Processing (NLP) applications (Jurgens and Pilehvar, 2015). It is general practice to identify and formalize conceptual relations using a reference knowledge repository. As such a repository, WordNet (Miller et al., 1990) stands out as the *de facto* standard lexical database, containing over 200k English senses with 155k word forms. Over the years, WordNet has become the cornerstone of agglutinative resources such as BabelNet (Navigli and Ponzetto, 2012) and Yago (Suchanek et al., 2007). It is also used in semantically intensive tasks such as Word Sense Disambiguation (Navigli, 2009), Query Expansion and IR (Fang, 2008), Sentiment Analysis (Esuli and Sebastiani, 2006), semantic similarity measurement (Pilehvar et al., 2013), development and evaluation of word embeddings models (Huang et al., 2012; Faruqui et al., 2015), and Taxonomy Learning Evaluation (Bordea et al., 2015).

While the value of WordNet for NLP is indisputable, it is generally recognized that enriching it with additional information makes it an even more valuable resource. Thus, there is a line of research aimed at extending it with novel terminology (Jurgens and Pilehvar, 2016), cross-predicate relations (Lopez de la Calle et al., 2016), and so forth. Nonetheless, one type of information has been largely neglected so far: collocations, i.e., idiosyncratic binary lexical co-occurrences. As a standalone research topic, however, collocations have been the focus of a substantial amount of work, e.g. for automatically retrieving them from corpora (Choueka, 1988; Church and Hanks, 1989; Smadja, 1993; Kilgariff, 2006; Evert, 2007; Pecina, 2008; Bouma, 2010; Gao, 2013), and for their semantic classification according to different typologies (Wanner et al., 2006; Gelbukh and Kolesnikova., 2012; Moreno et al., 2013; Wanner et al., 2016). However, to the best of our knowledge, no previous work attempted the automatic enrichment of WordNet with collocational information. The only related attempt consisted in designing a schema for

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 3422–3432, Osaka, Japan, December 11-17 2016.

the manual inclusion of lexical functions from Explanatory Combinatorial Lexicology (ECL) (Mel'čuk, 1996) into the Spanish EuroWordNet (Wanner et al., 2004).

Given the importance of collocations for a series of NLP applications (e.g. machine translation, text generation or paraphrasing), we propose to fill this gap by putting forward a new methodology which exploits intrinsic properties of state-of-the-art semantic vector space models and leverages the transformation matrix introduced by Mikolov et al. (2013b) in a word-level machine translation task. As a result, we release an extension of WordNet with detailed collocational information, named ColWordNet (CWN). This extension is carried out by means of the inclusion of *novel edges*, where each edge encodes a *collocates-with* relation, as well as the semantics of the collocation itself. For example, given the pair of synsets `desire.n.01` and `ardent.a.01`, a novel relation $\xrightarrow[x]{col:intense}$ is introduced, where 'intense' is the *semantic category* denoting *intensification*, and x is the confidence score assigned by our algorithm.

The remainder of the paper is organized as follows: In Section 2, we provide some background on collocations and the vector space models on which we base our approach. Section 3 describes the methodology followed to construct CWN. Then, Section 4 presents both intrinsic and extrinsic experimental results. And, finally, Section 5 summarizes the main contributions of our paper and outlines potential avenues for future work.

2 Background

In what follows, we first present relevant background on the semantic categories of collocations we use in our work (Section 2.1) and then on the resources used in our experiments (Section 2.2).

2.1 Collocations

Collocations are restricted lexical co-occurrences of two syntactically related lexical items, the base and the collocate. In a collocation, the base is freely chosen by the speaker, while the choice of the collocate depends on the base; see, e.g., (Cowie, 1994; Mel'čuk, 1996; Kilgariff, 2006) for a theoretical discussion. For instance, in the collocations *take* [*a*] *step*, *solve* [*a*] *problem*, *pay attention*, *deep sorrow*, and *strong tea*, *step*, *problem*, *attention*, *sorrow* and *tea* are the bases and *take*, *solve*, *pay*, *deep* and *strong* their respective collocates.

Besides a syntactic dependency, between the base and the collocate a semantic relation holds. Some of these semantic relations, such as 'intense', 'weak', 'perform', 'cause', etc. can be found across a large number of collocations. For instance, an 'intense' *applause* is a *thundering applause*, an 'intense' *emotion* is *deep*, 'intense' *rain* is *heavy*, and so on. In our experiments, we focused on the subset of the most prominent eight semantic collocation relations (or categories), which are listed in the first column of Table 1. These semantic categories are a generalization of the *lexical functions* (LFs) from ECL already used in Wanner et al. (2004). We have decided to use somewhat more generic categories instead of LFs because, on the one hand, some of the LFs differ only in terms of their syntactic structure (i.e. they capture the same semantic relation), and, on the other hand, LFs pose a great challenge for annotation due to their syntactic granularity.

2.2 Resources

The CWN lexical database is generated thanks to the exploitation of word and sense-based vector space models stemming from BabelNet (Navigli and Ponzetto, 2012),[1] which currently constitutes the largest semantic repository of both concepts and named entities.[2] In BabelNet, just like in WordNet, concepts are represented as *synsets* (i.e., set of synonym senses[3]). This allows us to exploit BabelNet's direct mapping with WordNet so that when our algorithm yields a candidate collocate $synset^n_{bn}$, we may retrieve

[1] `http://babelnet.org/`

[2] In its 3.6 release version, BabelNet is composed of 6.1M concepts and 7.7M named entities.

[3] For example, the concept defined as *principal activity in your life that you do to earn money* is represented by the synset {*occupation, business, job, line of work, line*}, where *occupation*, *business*, *job*, *line of work*, and *line* are senses/lexicalizations of the given synset.

its corresponding *synset*$^n_{wn}$, provided there exists one. In what follows, we briefly describe two different vector space models that are used in this paper for the task of synset-level collocation discovery.

SENSEMBED[4] (Iacobacci et al., 2015) is a knowledge-based approach for obtaining latent continuous representations of individual word senses based on Word2Vec (Mikolov et al., 2013a). Unlike other sense-based embeddings approaches, such as, e.g., Huang et al. (2012), which address the inherent polysemy of word-level representations relying solely on text corpora, SENSEMBED exploits the structured knowledge of BabelNet along with distributional information gathered from the Wikipedia corpus. In this paper, we used SENSEMBED for automatically disambiguating our training data, and as our **bases model**.

SHAREDEMBED. For this model we exploit distributional information from a 3B-word corpus extracted from the web (Han et al., 2013),[5] arguably richer in collocations than the encyclopedic style of Wikipedia. Similarly to SENSEMBED, this model is based on a pre-disambiguation of text corpora using BabelNet as sense inventory. However, unlike SENSEMBED, which learns vector representations for individual word senses, for this model we are interested in obtaining fine-grained information in the form of both plain text words and synsets[6] in a shared vector space (see Section 3.2 for the motivation behind this choice, and its application). To this end, we follow Chen et al. (2014) and modify the objective function of Word2Vec,[7] so that words and synsets can be learned jointly in a single training. The output is a vector space of word and synset embeddings that we use as **collocates model**.

3 Methodology

In this section, we provide a detailed description of the algorithm behind the construction of CWN. The system takes as input the WordNet lexical database and a set of collocation lists pertaining to predefined semantic categories, and outputs CWN. First, we collect training data and perform automatic disambiguation (Section 3.1). Then, we use this disambiguated data for training a linear *transformation matrix* from the base vector space, i.e., SENSEMBED, to the collocate vector space, i.e., SHAREDEMBED (Section 3.2). Finally, we exploit the WordNet taxonomy to select input base collocates to which we apply the transformation matrix in order to obtain a sorted list of candidate collocates (Section 3.3).

3.1 Collecting and Disambiguating Training Data

As is common in previous work on semantic collocation classification (Moreno et al., 2013; Wanner et al., 2016), our training set consists of a list of manually annotated collocations. For this purpose, we randomly selected nouns from the Macmillan Dictionary and manually classified their corresponding collocates with respect to their semantic categories.[8] Note that there may be more than one collocate for each base. Since collocations with different collocate meanings are not evenly distributed in language (e.g., we may tend to use more often collocations conveying the idea of 'intense' and 'perform' than 'begin to perform'), the number of instances per category in our training data also varies significantly (see Table 1).

Our training dataset consists at this stage of pairs of plain words, with the inherent ambiguity this gives raise to. We surmount this challenge by applying a disambiguation strategy based on the notion that, from all the available senses for a collocation's base and collocate, their correct senses are those which are most similar. This is a strategy that has been proved effective in previous concept-level disambiguation tasks (Delli Bovi et al., 2015). Formally, let us denote the SENSEMBED vector space as $\mathcal{S}$, and our original text-based training data as $\mathbf{T}$. For each training collocation $\langle b, c \rangle \in \mathbf{T}$ we consider all the available lexicalizations (i.e., senses) for both the base b and the collocate c in $\mathcal{S}$, namely $L_b = \{l_b^1 ... l_b^n\}$, and $L_c = \{l_c^1 ... l_c^m\}$, and their corresponding set of sense embeddings $\mathbf{V}_b = \{\vec{v}_b^1, ..., \vec{v}_b^n\}$ and $\mathbf{V}_c =$

[4] We downloaded the pre-trained sense embeddings at `http://lcl.uniroma1.it/sensembed/`.

[5] `ebiquity.umbc.edu/blogger/2013/05/01/umbc-webbase-corpus-of-3b-english-words/`

[6] As explained above, a synset is a set composed of synonym senses.

[7] We used the Continous Bag-Of-Words (CBOW) model with standard hyperparameters: 300 dimensions and a window size of 8 words.

[8] We do not consider phrasal verb collocates, e.g. *stand up*, *give up* or *calm down*.

Sem. Category	Example	# instances
‘intense’	*absolute certainty*	586
‘weak’	*remote chance*	70
‘perform’	*give chase*	393
‘begin to perform’	*take up a chase*	79
‘increase’	*improve concentration*	73
‘decrease’	*limit* [*a*] *choice*	73
‘create’, ‘cause’	*pose* [*a*] *challenge*	195
‘put an end’	*break* [*the*] *calm*	79

Table 1: Semantic categories and size of training set

$\{\vec{v}_c^1, ..., \vec{v}_c^m\}$. Our aim is to select, among all possible pairs of senses, the pair $\langle l'_b, l'_c\rangle$ that maximizes the cosine similarity between the corresponding embeddings v'_b and v'_c, which is computed as follows:

$$\langle \vec{v}\,'_b, \vec{v}\,'_c\rangle = \mathrm{argmax}_{\vec{v}_b \in \mathbf{V}_b,\, \vec{v}_c \in \mathbf{V}_c} \frac{\vec{v}_b \cdot \vec{v}_c}{\|\vec{v}_b\|\,\|\vec{v}_c\|} \tag{1}$$

Our disambiguation strategy yields a set of disambiguated pairs **D**. This is the input for the next module of the pipeline, the learning of a *transformation matrix* aimed at retrieving WordNet synset collocates for any given WordNet synset base.

3.2 Training a Sense-Level Transformation Matrix for each Semantic Category

Among the many properties of word embeddings (Mikolov et al., 2013a; Mikolov et al., 2013c) that have been explored so far in the literature (e.g., modeling analogies or projecting similar words nearby in the vector space), the most pertinent to this work is the linear relation that holds between semantically similar words in two analogous spaces (Mikolov et al., 2013b). Mikolov et al.’s original work learned a linear projection between two monolingual embeddings models to train a word-level machine translation system between English and Spanish. Other examples include the exploitation of this property for language normalization, i.e. finding *regular English* counterparts of Twitter language (Tan et al., 2015), or hypernym discovery (Espinosa-Anke et al., 2016).

In our specific case, we learn a linear transformation from $\vec{v}\,'_b$ to $\vec{v}\,'_c$, aiming at reflecting an inherent condition of collocations. Since collocations are a linguistic phenomenon that is more frequent in the narrative discourse than in formal essays, they are less likely to appear in an encyclopedic corpus (recall that SENSEMBED vectors, which we use, are trained on a dump of the English Wikipedia). This motivates the use of $\mathcal{S}$ as our *base space*, and our SHAREDEMBED $\mathcal{X}$ as the *collocate model*, as it was trained over more varied language such as blog posts or news items.

Then, we construct our linear transformation model as follows: For each disambiguated collocation $\langle l'_b, l'_c\rangle \in \mathbf{D}$, we first retrieve the corresponding base vectors $\vec{v}\,'_b$. Next, we exploit the fact that $\mathcal{X}$ contains both BabelNet synsets and words, and derive for each l'_c two items, namely the vectors associated to its lexicalization (word-based) and its BabelNet synset. For example, for the training pair ⟨`ardent_bn:00097467a,desire_bn:00026551n`⟩ $\in$ **D**, we learn two linear mappings, namely `ardent_bn:00097467a` ↦ `desire` and `ardent_bn:00097467a` ↦ `bn:00026551n`. We opt for this strategy, which doubles the size of the training data in most lexical functions (depending on coverage), due to the lack of resources of manually-encoded classification of collocations. By following this strategy we obtain an extended training set $\mathbf{D}{*} = \{\vec{b}_i, \vec{c}_i\}_{i=1}^n$ ($\vec{b}_i \in \mathcal{X}$, $\vec{c}_i \in \mathcal{S}, n \geq |\mathbf{D}|$). Then, we construct a *base matrix* $\mathbf{B} = \left[\vec{b}_1 \ldots \vec{b}_n\right]$ and a collocate matrix $\mathbf{C} = [\vec{c}_1 \ldots \vec{c}_n]$ with the resulting set of training vector pairs. We use these matrices to learn a linear transformation matrix $\Psi \in \mathbb{R}^{d_\mathcal{S} \times d_\mathcal{X}}$, where $d_\mathcal{S}$ and and $d_\mathcal{X}$ are, respectively, the number of dimensions of the base vector space (i.e., SENSEMBED) and the collocate vector space (SHAREDEMBED).[9] Following the notation in Tan et al. (2015), this transformation can be depicted as:

[9] In our setting the numbers of dimensions are $d_\mathcal{S} = 400$ and $d_\mathcal{X} = 300$.

$$\mathbf{B}\Psi \approx \mathbf{C}$$

As in Mikolov et al.'s original approach, the training matrix is learned by solving the following optimization problem:

$$\min_{\Psi} \sum_{i=1}^{n} \|\vec{b}_i - \vec{c}_i\|^2$$

Having trained Ψ, the next step of the pipeline is to apply it over a subset of WordNet's base concepts and their hyponyms. For each synset in this branch, we apply a scoring and ranking procedure which assigns a *collocates-with* score. If such score is higher than a predefined threshold, tuned over a development set, this relation is included in CWN.

3.3 Retrieving and Sorting WordNet Collocate Synsets

During the task of enriching WordNet with collocational information, we first gather a set of base WordNet synsets by traversing WordNet hypernym hierarchy starting from those base concepts that are most fit for the input semantic category.[10] Then, the *transformation matrix* learned in Section 3.2 is used to find candidate WordNet synset collocates (mostly verbs or adjectives) for each base WordNet synset.

As explained in Section 3, WordNet synsets are mapped to BabelNet synsets, which in turn map to as many vectors in SENSEMBED as their associated lexicalizations. Formally, given a base synset b, we apply the transformation matrix to all the SENSEMBED vectors $\mathbf{V}_b = \{\vec{v}_b^{\,1}, ..., \vec{v}_b^{\,n}\}$ associated with its lexicalizations. For each $\vec{v}_b^{\,i} \in \mathbf{V}_b$, we first get the vector $\vec{\psi}_b^{\,i} = \vec{v}_b^{\,i}\Psi$ obtained as a result of applying the transformation matrix and then we gather the subset $W_b^i = \{\vec{w}_b^{\,i,1} \ldots \vec{w}_b^{\,i,10}\}$ ($\vec{w}_b^{\,i,j} \in \mathcal{X}$) of the top ten closest vectors by cosine similarity to $\vec{\psi}_b^{\,i}$ in the SHAREDEMBED vector space $\mathcal{X}$. Each $\vec{w}_b^{\,i,j}$ is ranked according to a scoring function $\lambda(\cdot)$, which is computed as follows[11]: $\lambda(\vec{w}_b^{\,i,j}) = \frac{cos(\vec{\psi}_b^{\,i}, \vec{w}_b^{\,i,j})}{j}$. This scoring function takes into account both the cosine similarity as well as the relative position[12] of the candidate collocate with respect to other neighbors in the vector space. Apart from sorting the list of candidate collocates, this scoring function is also used to measure the confidence of the retrieved collocate synsets in CWN.

4 Evaluation

We evaluate CWN both intrinsically and extrinsically. Our intrinsic evaluation consists of a manual scoring of the correctness of the newly introduced relations (Section 4.1). Extrinsic evaluation assesses the quality of CWN as an input resource for introducing collocational information into a word embeddings model (Section 4.2).

4.1 Intrinsic: Precision of Collocate Relations

Sampling and evaluation are carried out as follows. First, for each semantic category, we retrieve 50 random bases included in the aforementioned base concepts (see Section 3.3) and all their hyponym branch. This results in an evaluation set $Test$ of 800 collocations, as for each base we retrieve the 5 highest scoring candiates. These collocations are evaluated in terms of correctness, i.e., if the associated synset is an appropriate collocate for the input base. Note that not all bases in the test set may be suitable for the given semantic category, and that is why we also perform an evaluation on the test data restricted to only those bases manually selected for being suitable for having at least one collocate. We denote the restricted test data as $Test*$. For example, the base synset `putt.n.01` defined as *hitting a golf ball*

[10] These are: For 'intense' and 'weak', `attitude.n.01`, `feeling.n.01` and `ability.n.02`. For the rest of them, we select `cognition.n.01`, `act.n.02` and `action.n.01`.

[11] If $w_b^{i,j}$ appears in a different W_b^j set ($j \neq i$), its scores are averaged.

[12] Position is arguably an important factor as there may be dense areas where cosine similarity alone may not reflect entirely the fitness of a candidate.

	'intense'				**'perform'**				**'put an end'**				**'increase'**			
	Baseline		CWN		Baseline		CWN		Baseline		CWN		Baseline		CWN	
	Test	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**
P@1	0.00	0.00	0.35	**0.46**	0.15	0.16	0.20	**0.36**	0.05	0.08	0.15	**0.50**	0.05	0.14	0.15	**0.42**
P@5	0.03	0.30	0.43	**0.57**	0.06	0.06	0.13	**0.23**	0.02	0.03	0.12	**0.40**	0.04	0.11	0.18	**0.51**
MRR	0.05	0.41	0.48	**0.65**	0.18	0.19	0.32	**0.59**	0.07	0.12	0.20	**0.68**	0.07	0.21	0.22	**0.65**
MAP	0.05	0.45	0.48	**0.64**	0.15	0.18	0.32	**0.59**	0.07	0.12	0.19	**0.64**	0.07	0.20	0.22	**0.64**
	'decrease'				**'create/cause'**				**'weak'**				**'begin to perform'**			
	Baseline		CWN		Baseline		CWN		Baseline		CWN		Baseline		CWN	
	Test	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**
P@1	0.00	0.00	0.30	**0.46**	0.05	0.16	0.10	**0.50**	0.00	0.00	0.10	**0.22**	0.00	0.00	0.00	0.00
P@5	0.02	0.03	0.19	**0.29**	0.04	0.13	0.04	**0.20**	0.02	0.03	0.04	**0.08**	0.03	0.07	0.02	**0.20**
MRR	0.02	0.04	0.39	**0.61**	0.07	0.25	0.10	**0.50**	0.03	0.04	0.01	**0.22**	0.05	0.12	0.04	**0.41**
MAP	0.02	0.03	0.38	**0.58**	0.06	0.20	0.10	**0.50**	0.03	0.04	0.01	**0.22**	0.05	0.12	0.04	**0.41**

Table 2: Summary of the manual evaluation of the performance of CWN and of the baseline

that is on the green using a putter does not admit any 'decrease' collocate, and therefore its collocations are not considered in $Test*$.

Since our algorithm returns a list of candidate collocate synsets for an input base synset, the task naturally becomes that of a ranking problem, and therefore ranking metrics such as Precision@K (P@K), Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR) are appropriate for evaluating this experiment. These measures provide insights on different aspects of the outcome of the task, e.g. how often valid collocates were retrieved in the first positions of the rank (MRR), and if there were more than one valid collocate, whether this set was correctly retrieved, (MAP and R-P)[13]. In Table 2 we provide a detailed summary of the performance of our system (CWN), as compared with a competitor unsupervised baseline which exploits word analogies (as in $\vec{man} - \vec{king} + \vec{woman} = \vec{queen}$). This baseline, which we deploy on the SHAREDEMBED space, takes as input a prototypical collocation of a given semantic category (e.g. *thunderous applause* for 'intense') and an input base, and collects the top 10 Nearest Neighbours (NNs) to the vector resulting of the aforementioned analogy operation. This approach was recently used in a similar setting (Rodríguez-Fernández et al., 2016). Due to the difficulty of the task, and the restriction it imposes for collocates to be disambiguated synsets rather than any text-based word, the unsupervised approach fails short when compared to our supervised method, which is capable to find more and better disambiguated collocates.

Note that for half of the semantic categories under evaluation, our approach correlated well with human judgement, with the highest ranking candidates being more often correct than those ranked lower. This is the case of 'put an end', 'decrease', 'create/cause' and 'weak'. In fact, it is in 'put an end', where our system achieves the highest MRR score, which we claim to be the most relevant measure, as it rewards cases where the first ranked returned collocation is correct without measuring in the retrieved collocates at other positions. Moreover, let us highlight the importance of two main factors. First, the need for a well-defined semantic relation between bases and collocates. It has been shown in other tasks that exploit linear transformations between embeddings models that even for one single relation there may be clusters that require certain specificity in the *domain* or *semantic* of the data (see Fu et al. Fu et al. (2014) for a discussion of this phenomenon in the task of taxonomy learning). Second, the importance of having a reasonable amount of training pairs so that the model can learn the idiosyncrasies of the semantic relation that is being encoded (e.g., Mikolov et al. (2013b) report a major increase in performance as training data increases in several orders of magnitude). This is reinforced in our experiments, where we obtain the highest MAP score for 'intense', the semantic category for which we have the largest training data available.

[13] See Bian et al. (2008) for an in-depth analysis of these metrics.

	'intense'			'weak'			'perform'			'create/cause'		
	correct	*dist.*	*diff.*	*correct*	dist.	*diff.*	*correct*	*dist.*	*diff.*	*correct*	*dist.*	*diff.*
original	0.22	0.04	+0.18	0.17	0.05	+0.12	0.15	0.05	+0.10	0.17	0.06	+0.11
retrofitted	0.27	0.06	**+0.21**	0.19	0.06	**+0.13**	0.25	0.11	**+0.14**	0.28	0.12	**+0.16**

Table 3: Comparison of collocational sensitivity between original and retrofitted embeddings models over four semantic categories.

4.2 Extrinsic evaluation: Retrofitting Vector Space Models to CWN

We complement our manual evaluation with an extrinsic experiment, where we assess the extent to which our newly generated lexical resource can be used to *introduce collocational sensitivity* to a generic word embeddings model.[14] To this end, we extract collocation clusters by extracting all the synsets associated lemmas (e.g. for *heavy.a.01 rain.n.01*, we would extract the cluster [*heavy*, *rain*, *rainfall*]). These are used as input for the Retrofitting Word Vectors algorithm (Faruqui et al., 2015).[15] This algorithm takes as input a vector space and a semantic lexicon which may encode any semantic relation, and puts closer in the vector space words that are related in the lexicon.

Previous approaches have encoded semantic relations by introducing some kind of bias into a vector space model (Yu et al., 2015; Pham et al., 2015; Mrkšić et al., 2016; Nguyen et al., 2016). For instance, Yu et al. (2015) encode (term, hypernym) relations by grouping together terms and their hypernyms, rather than semantically related items. In this way, their *biased* model puts closer to *jaguar* terms like *animal* or *vehicle*, while an unbiased model would put nearby terms such as *lion*, *bmw* or *jungle*. We aim at introducing a similar bias, but in terms of collocational information. This is achieved, for each lexical function and each synset in CWN-*st*, by obtaining its top 3 collocate candidates and incorporate information on their *collocationality* into the model.

4.2.1 Collocational Sensitivity

In this experiment, we assess the extent to which a retrofitted model with collocational bias is able to discriminate between a correct collocation and a random combination of the same base with an unrelated collocate. To this end, we manually constructed two datasets, one for *noun+adjective* ('intense' and 'weak' semantic categories) and one for *noun+verb* combinations, which we evaluate on the two most productive semantic categories, namely 'perform' and 'create/cause'. The datasets consist of 50 bases and one of their correct collocates according to the Macmillan Collocations Dictionary, accompanied by four *distractor* (*dist.* in Table 3) collocates. For instance, given the correct 'perform' collocation *make a pledge*, we expect our 'perform'-wise retrofitted model to increase the score in $\vec{make} + \vec{pledge}$ substantially more than a combination $\vec{pledge} + \vec{distractor}$. For each evaluated semantic category, we computed the average increase of the cosine similarity between all correct collocations and all distractors (*diff.* in Table 3). As shown in Table 3, there is a consistent increase over the four evaluated semantic categories, namely 'intense', 'weak', 'perform' and 'create/cause'. This proves the potential of our retrofitted model to discern between correct and wrong collocates. In the following section, we explore the possibility to use this vector space for finding collocates giving a base as input.

4.2.2 Exploring Nearest Neighbours for Collocate Discovery

Inspired by Yu et al.'s (2015) work on introducing hypernymic bias into a word embeddings model, we explore the extent to which our retrofitted models can be used to discover *alternative collocates* given the composition of the words involved in a collocation as input. In order to discover these collocates, we compose the base and the collocate by averaging their respective word embeddings and retrieve its closest words in the vector space according to cosine similarity. In Table 4, we show a sample of five NNs for several input adjective+noun collocations. These examples reveal how the vector space model retroffited using our collocations tends to bring closer in the space modifiers (i.e., collocates), providing

[14] We use the Google News pre-trained Word2Vec vectors, available at `code.google.com/archive/p/word2vec/`, as input for retrofitting.

[15] We used the code available at `https://github.com/mfaruqui/retrofitting`

	‘**intense**’			‘**weak**’	
	`original`	`retrofitted`		`original`	`retrofitted`
ferocious + hatred	**vicious**	fierce	dim +light	bright	**faint**
	fury	**fearsome**		dimmed	**unaccented**
	ferocity	fury		dimmer	**dense**
	savage	hate		dimming	bright
	hostility	**savage**		lights	centaur
intense + sympathy	fierce	**considerable**	mild + comment	milder	**modest**
	empathy	**tremendous**		`NamedEntity`	**meek**
	admiration	**enormous**		`NamedEntity`	`NamedEntity`
	anger	encouragement		`NamedEntity`	`NamedEntity`
	grudging respect	**immense**		`NamedEntity`	`NamedEntity`
sheer + delight	amazement	**immense**	modest + progress	progress	**mild**
	sheer unadulterated	**colossal**		pro gress	**meek**
	sheer joy	delectation		**Modest**	dissatisfaction
	joy	disgust		`NamedEntity`	progess
	astonishment	**stupendous**		strides	**slight**

Table 4: Comparison of the five NNs of six sample adj+noun collocations between a generic word embeddings model and a *retrofitted* version with semantic collocation information (‘intense’ and ‘weak’). Note the increase in plausible collocates in retrofitted models (in bold). `NamedEntity` refers to noisy entities appearing among the top 5 NNs.

an interesting method for automatic collocation discovery. Despite its simplicity, this collocational discovery approach extracts a considerable amount of suitable fine-grained collocates for a given base. For example, given the collocation *intense sympathy*, the retrofitted space extracts *considerable*, *tremendous*, *enormous* and *immense* as candidate collocates of intensity among the five nearest neighbours. As future work we plan to further exploit and evaluate the impact of this property.

5 Conclusions and Future Work

We have described a system for an automatic enrichment of the WordNet lexical database with fine-grained collocational information, yielding a resource called ColWordNet (CWN). Our approach is based on the intuition that there is a linear transformation in vector spaces between bases and collocates of the same *semantic category*, e.g. between *heavy* and *rain*, or between *ardent* and *desire*. We have exploited sense-based embedding models to train an algorithm designed to retrieve valid collocates for a given input base. This pipeline is carried out at the *sense* level (rather than the word level), by leveraging models which use BabelNet as a reference sense inventory. We evaluated CWN both intrinsically and extrinsically, and verified that our algorithm is able to encode fine-grained *collocates-with* relations at synset level.

Release. We release CWN at several different confidence levels. The version with the highest confidence includes over 100k collocational edges, which connect over 8k unique base and collocate WordNet synsets. These connections are further enriched by two pieces of information, namely (1) the type of collocation (e.g. ‘intense’ or ‘perform’), and (2) a confidence score derived from our approach. Moreover, in addition to CWN, we also release four modified versions of the well-known Word2Vec Google News vector space model, retrofitted with collocational information, which we constructed for the extrinsic evaluation of CWN. These models can be exploited both for assessing the correctness of a collocation and for the discovery of alternative collocates for a given collocation. Finally, we also make available the evaluation datasets built as part of the Collocational Sensitivity experiment. All data associated with this publication is publicly available at `http://www.taln.upf.edu/colwordnet`.

Future work. In the future, we plan to design a method to retrieve the best *bases* for a given semantic category, which would allow us not to rely on predefined manually built base concepts. Finally, we are currently investigating the potential of applying neural approaches recasting the task as a sequence classification problem for including collocational information in WordNet clusters.

Acknowledgements

This work was partially funded by the following projects: MULTISENSOR (FP7-ICT-610411), KRISTINA (H2020-645012-RIA), HARENES (FFI201130219-CO2-02), through a predoctoral grant (BES-2012-057036), the María de Maeztu Units of Excellence Program (MDM-2015-0502) and TUNER (TIN2015-65308-C5-5-R, MINECO/FEDER, UE). Jose Camacho-Collados is supported by a Google Doctoral Fellowship in Natural Language Processing.

References

Jiang Bian, Yandong Liu, Eugene Agichtein, and Hongyuan Zha. 2008. Finding the right facts in the crowd: factoid question answering over social media. In *Proceedings of the 17th international conference on World Wide Web*, pages 467–476. ACM.

Georgeta Bordea, Paul Buitelaar, Stefano Faralli, and Roberto Navigli. 2015. Semeval-2015 task 17: Taxonomy extraction evaluation (texeval). In *Proceedings of the SemEval workshop*.

G. Bouma. 2010. Collocation Extraction beyond the Independence Assumption. In *Proceedings of ACL, Short paper track*, pages 109–114, Uppsala, Sweden.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In *Proceedings of EMNLP*, pages 1025–1035, Doha, Qatar.

Y. Choueka. 1988. Looking for Needles in a Haystack or Locating Interesting Collocational Expressions in Large Textual Databases. In *Proceedings of the RIAO*, pages 34–38.

K. Church and P. Hanks. 1989. Word Association Norms, Mutual Information, and Lexicography. In *Proceedings of ACL*, pages 76–83.

A. Cowie. 1994. Phraseology. In R.E. Asher and J.M.Y. Simpson, editors, *The Encyclopedia of Language and Linguistics, Vol. 6*, pages 3168–3171. Pergamon, Oxford.

Claudio Delli Bovi, Luis Espinosa Anke, and Roberto Navigli. 2015. Knowledge base unification via sense embeddings and disambiguation. In *Proceedings of EMNLP*, pages 726–736, Lisbon, Portugal.

Luis Espinosa-Anke, Jose Camacho-Collados, Claudio Delli Bovi, and Horacio Saggion. 2016. Supervised distributional hypernym discovery via domain adaptation. In *Proceedings of EMNLP*.

Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, volume 6, pages 417–422.

S. Evert. 2007. Corpora and Collocations. In A. Lüdeling and M. Kytö, editors, *Corpus Linguistics. An International Handbook*. Mouton de Gruyter, Berlin.

Hui Fang. 2008. A re-examination of query expansion using lexical resources. In *ACL*, volume 2008, pages 139–147.

Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of NAACL*, pages 1606–1615.

Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning semantic hierarchies via word embeddings. In *Proceedings of ACL*, volume 1.

Z.M. Gao. 2013. Automatic Identification of English Collocation Errors based on Dependency Relations. *Sponsors: National Science Council, Executive Yuan, ROC Institute of Linguistics, Academia Sinica NCCU Office of Research and Development*, page 550.

A. Gelbukh and O. Kolesnikova. 2012. *Semantic Analysis of Verbal Collocations with Lexical Functions*. Springer, Heidelberg.

Lushan Han, Abhay Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. 2013. UMBC EBIQUITY-CORE: Semantic textual similarity systems. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics*, volume 1, pages 44–52.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of ACL*, pages 873–882, Jeju Island, Korea.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. SensEmbed: Learning Sense Embeddings for Word and Relational Similarity. In *Proceedings of ACL*, pages 95–105, Beijing, China.

David Jurgens and Mohammad Taher Pilehvar. 2015. Reserating the awesometastic: An automatic extension of the wordnet taxonomy for novel terms. In *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: Human language technologies, Denver, CO*, pages 1459–1465.

David Jurgens and Mohammad Taher Pilehvar. 2016. SemEval-2016 Task 14: Semantic taxonomy enrichment. *Proceedings of SemEval*, pages 1092–1102.

Adam Kilgariff. 2006. Collocationality (and How to Measure it). In *Proceedings of the Euralex Conference*, pages 997–1004, Turin, Italy. Springer-Verlag.

Maddalen Lopez de la Calle, Itziar Aldabe, Egoitz Laparra, and German Rigau. 2016. Predicate Matrix. Atomatically extending the semantic interoperability between predicate resources. *Language Resources and Evaluation*, 50(2):263–289.

I.A. Mel'čuk. 1996. Lexical Functions: A tool for the description of lexical relations in the lexicon. In L. Wanner, editor, *Lexical Functions in Lexicography and Natural Language Processing*, pages 37–102. Benjamins Academic Publishers, Amsterdam.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013b. Exploiting similarities among languages for Machine Translation. *arXiv preprint arXiv:1309.4168*.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751.

George A. Miller, R.T. Beckwith, Christiane D. Fellbaum, D. Gross, and K. Miller. 1990. WordNet: an online lexical database. *International Journal of Lexicography*, 3(4):235–244.

Pol Moreno, Gabriela Ferraro, and Leo Wanner. 2013. Can we determine the semantics of collocations without using semantics? In I. Kosem, J. Kallas, P. Gantar, S. Krek, M. Langemets, and M. Tuulik, editors, *Proceedings of the eLex 2013 conference*, Tallinn & Ljubljana. Trojina, Institute for Applied Slovene Studies & Eesti Keele Instituut.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. *arXiv preprint arXiv:1603.00892*.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. *Artificial Intelligence*, 193:217–250.

Roberto Navigli. 2009. Word Sense Disambiguation: A Survey. *ACM Computing Surveys (CSUR)*, 41(2):10.

Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. 2016. Integrating Distributional Lexical Contrast into Word Embeddings for Antonym-Synonym Distinction. In *Proc. of ACL*, pages 454–459.

Pavel. Pecina. 2008. A Machine Learning Approach to Multiword Expression Extraction. In *Proceedings of the LREC 2008 Workshop Towards a Shared Task for Multiword Expressions (MWE 2008)*, pages 54–57, Marrakech.

Nghia The Pham, Angeliki Lazaridou, and Marco Baroni. 2015. A Multitask Objective to Inject Lexical Contrast into Distributional Semantics. In *Proceedings of ACL*, pages 21–26.

Mohammad Taher Pilehvar, David Jurgens, and Roberto Navigli. 2013. Align, Disambiguate and Walk: A Unified Approach for Measuring Semantic Similarity. In *Proceedings of ACL*, pages 1341–1351.

Sara Rodríguez-Fernández, Roberto Carlini, Luis Espinosa-Anke, and Leo Wanner. 2016. Example-based Acquisition of Fine-grained Collocation Resources. In *Proceedings of LREC*, Portoroz, Slovenia.

Frank Smadja. 1993. Retrieving Collocations from Text: X-Tract. *Computational Linguistics*, 19(1):143–177.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A Core of Semantic Knowledge. In *WWW*, pages 697–706. ACM.

Luchen Tan, Haotian Zhang, Charles L.A. Clarke, and Mark D. Smucker. 2015. Lexical Comparison Between Wikipedia and Twitter Corpora by Using Word Embeddings. In *Proceedings of ACL (2)*, pages 657–661, Beijing, China, July.

Leo Wanner, Margarita Alonso Ramos, and Antonia Martí. 2004. Enriching the Spanish EuroWordNet by Collocations. In *LREC*.

Leo Wanner, Bernd Bohnet, and Mark Giereth. 2006. Making Sense of Collocations. *Computer Speech and Language*, 20(4):609–624.

Leo Wanner, Gabriela Ferraro, and Pol Moreno. 2016. Towards Distributional Semantics-based Classification of Collocations for Collocation Dictionaries. *International Journal of Lexicography, doi:10.1093/ijl/ecw002*.

Zheng Yu, Haixun Wang, Xuemin Lin, and Min Wang. 2015. Learning Term Embeddings for Hypernymy Identification. In *Proceedings of IJCAI*, pages 1390–1397.