NASARI: Integrating Explicit Knowledge and Corpus Statistics for a Multilingual Representation of Concepts and Entities

José Camacho-Collados, Mohammad Taher Pilehvar[a,1], Roberto Navigli

Department of Computer Science
Sapienza University of Rome

[a]*Department of Theoretical and Applied Linguistics*
University of Cambridge

Abstract

Owing to the need for a deep understanding of linguistic items, semantic representation is considered to be one of the fundamental components of several applications in Natural Language Processing and Artificial Intelligence. As a result, semantic representation has been one of the prominent research areas in lexical semantics over the past decades. However, due mainly to the lack of large sense-annotated corpora, most existing representation techniques are limited to the lexical level and thus cannot be effectively applied to individual word senses. In this paper we put forward a novel multilingual vector representation, called NASARI, which not only enables accurate representation of word senses in different languages, but it also provides two main advantages over existing approaches: (1) high coverage, including both concepts and named entities, (2) comparability across languages and linguistic levels (i.e., words, senses and concepts), thanks to the representation of linguistic items in a single unified semantic space and in a joint embedded space, respectively. Moreover, our representations are flexible, can be applied to multiple applications and are freely available at `http://lcl.uniroma1.it/nasari/`. As evaluation benchmark, we opted for four different tasks, namely, word similarity, sense clustering, domain labeling, and Word Sense Disambiguation, for each of which we report state-of-the-art performance on several standard datasets across different languages.

Keywords:
semantic representation, lexical semantics, Word Sense Disambiguation, semantic similarity, sense clustering, domain labeling

1. Introduction

Semantic representation, i.e., modeling the semantics of a linguistic item[2] in a mathematical or machine interpretable form, is a fundamental problem in Natural Language Processing (NLP) and Artificial Intelligence (AI). Because they represent the lowest linguistic level, word senses play a vital role in natural language understanding. Effective representations of word senses can be directly useful to Word Sense Disambiguation [94], semantic similarity [13, 130, 107], coarsening sense inventories [93, 125], alignment of lexical resources [102, 99, 109], lexical substitution [75], and semantic priming [101]. Moreover, sense-level representation can be directly extended to applications requiring word representations, with the added benefit that it provides extra semantic information. Turney and Pantel [130] provide a review of some of the applications of word representation, including: automatic thesaurus generation [21, 22], word similarity [25, 129, 114] and clustering [104], query expansion [141], information extraction [61], semantic role labeling [29, 105], spelling correction [53], and Word Sense Disambiguation [94].

[1]Work mainly done at the Sapienza University of Rome.

[2]Throughout this article by a linguistic item we mean any kind of linguistic unit that can bear a meaning, i.e., a word sense, a word, a phrase, a sentence or a larger piece of text.

The Vector Space Model (VSM) is a prominent approach for semantic representation. The model represents a linguistic item as a vector (or a point) in an n-dimensional semantic space, i.e., a mathematical space wherein each of the n dimensions (hence, axes of the space) denotes a single linguistic entity, such as a word. The popularity of the VSM representation is due to two main reasons. Firstly, it is straightforward to view vectors as sets of features and directly apply various machine learning techniques on them. Secondly, the model enjoys support from the field of Cognitive Science wherein several studies have empirically or theoretically suggested that various aspects of human cognition accord with VSMs [36, 64].

However, most VSM-based techniques, whether in their conventional co-occurrence based form [120, 130, 63], or in their newer predictive branch [20, 82, 8], usually base their computation on the distributional statistics derived from text corpora. Hence, in order to be able to represent individual meanings of words (i.e., word senses), these techniques require large amounts of disambiguated text prior to modeling. Additionally, Word Sense Induction techniques [104, 11, 58, 27] require sense-annotated data, if their induced sense clusters are to be mapped to an existing sense inventory. However, providing sense-annotated data on a large scale is a time-consuming process which has to be carried out separately for each word sense and repeated for each new language of interest, i.e., the so-called knowledge acquisition bottleneck. Importantly, the largest manual effort for providing a wide-coverage sense-annotated dataset dates back to 1993, in the case of the SemCor corpus [86]. In fact, although cheap and fast annotations could be obtained by means of Amazon Mechanical Turk [124, 55], games with a purpose [134, 132, 56], or voluntary collaborative editing such as in Wikipedia [77], producing annotated resources manually is still an onerous task. On the other hand, the performance of Word Sense Disambiguation (WSD) techniques is still far from ideal [94], which in its turn prevents a reliable automatic sense-annotation of large text corpora that can be used for modeling individual word senses. This hinders the functionality of this group of vector space models in tasks such as Word Sense Disambiguation (WSD) that require the representation of individual word senses.

There have been several efforts to adapt and apply distributional approaches to the representation of word senses [104, 12, 115, 47, 68]. However, most of these techniques cannot provide representations that are already linked to a standard sense inventory, and consequently such mapping has to be carried out either manually, or with the help of sense-annotated data [48]. Recently, there have been attempts to address this issue and to obtain vectors for individual word senses by exploiting the WordNet semantic network [74, 107, 109, 117] and its glosses [19]. These approaches, however, are either restricted to the representation of concepts defined in WordNet and to the English language only, or are designed for specific tasks.

In our recent work [16], we proposed a method that exploits the structural knowledge derived from semantic networks, together with distributional statistics from text corpora, to produce effective representations of individual word senses or concepts. Our approach provides two main advantages in comparison to previous VSM techniques. Firstly, it is multilingual, as it can be directly applied for the representation of concepts in dozens of languages. Secondly, each vector represents a concept, irrespective of its language, in a unified semantic space having concepts as its dimensions, permitting direct comparison of different representations across languages and hence enabling cross-lingual applications.

In this article, we improve our approach, referred to as NASARI (Novel Approach to a Semantically-Aware Representation of Items) henceforth, and extend their application to a wider range of tasks in lexical semantics. Specifically, the novel contributions are as follow:

1. We propose a new formulation for fast computation of lexical specificity (Section 3.1.1).
2. We propose a new flexible way to get continuous embedded vector representations, with the added benefit of obtaining a semantic space shared by BabelNet synsets, words and texts (Section 3.3).
3. We put forward a technique for improved computation of weights in the unified vectors and show how it can improve the accuracy and efficiency of the representations (Section 3.4).
4. We compute and assign weights to individual edges in our semantic network (Section 4.1) and show by means of different experiments the advantage we gain when using this new weighted graph (Section 10).
5. We release the lexical and unified vector representations for five different languages (English, French, German, Italian and Spanish) and the embedded vector representations for the English language at `http://lcl.uniroma1.it/nasari/` .

In addition to these contributions, we also devised robust frameworks that enable direct application of our representations to four different tasks: Semantic Similarity (Section 6), Sense Clustering (Section 7), Domain Labeling

(Section 8) and Word Sense Disambiguation (Section 9). For each of the tasks, we carried out a comprehensive set of evaluations on several datasets in order to verify the reliability and flexibility of NASARI different datasets and tasks. We provide a summary of the experiments in Section 5.

The rest of this article is structured as follows. We first provide an introduction of some of the most widely used knowledge resources in lexical semantics, in Section 2. After which, in Section 3 we describe our methodology to convert text into lexical, embedded and unified vectors. The process to obtain vector representations for synset vectors by leveraging the knowledge resources described in Section 2, and the methodology to obtain vectors from text described in Section 3, is presented in Section 4. We present a summary of the experiments and the performance of NASARI across tasks in Section 5. Then, we describe some applications of the vectors with their respective frameworks and experiments in Sections 6 (Semantic Similarity), 7 (Sense Clustering), 8 (Domain Labeling), and 9 (Word Sense Disambiguation). We analyze the performance of different components of our model in Section 10. Finally, we discuss the related work in Section 11 and provide the concluding remarks in Section 12.

2. Knowledge Resources

Knowledge resources can be divided into two general categories: expert made and collaboratively constructed. Each type has its own advantages and limits. Manually-annotated resources feature highly-accurate encoding of concepts and semantic relationships between them but, with a few exceptions, are usually limited in their lexical coverage, and are typically focused on a specific language only. A good example is **WORDNET** [84], a semantic network whose basic units are synsets. A synset represents a concept which may be expressed through nouns, verbs, adjectives or adverbs and is composed of the different lexicalizations (i.e., synonyms that are used to express it). For example, the synset of the *middle of the day* concept comprises six lexicalizations: *noon, twelve noon, high noon, midday, noonday, noontide*. Synsets may also be seen as nodes in a semantic network. These nodes are connected to each other by means of lexical or semantic relations (hypernymy, meronymy, etc.). These relations are seen as the edges in the WordNet semantic network. Despite being one of the largest and most complete manually-made lexical resources, WordNet still lacks coverage of lemmas and senses from domain specific lexicons (e.g., law or medicine), named entities, creative slang usages, or those for technology that came into existence only recently.

On the other hand, collaboratively-constructed resources, such as **WIKIPEDIA**, provide features such as multilinguality, wide coverage and up-to-dateness. As of September 2015, Wikipedia provides more than 100K articles in over fifty languages. This coverage is steadily increasing. For instance, the English Wikipedia alone receives 750 new articles per day. Each of these articles provides, for its corresponding concept, a great deal of information in the form textual information, tables, infoboxes, and various relations (such as redirections, disambiguations, and categories). These features have persuaded many researchers over the past few years to exploit the huge amounts of semi-structured knowledge available in such collaborative resources for different NLP applications [46, 126].

The types of knowledge available in the expert-based and collaboratively-constructed resources make them complementary. This has motivated researchers to combine various lexical resources across the two categories [102, 109]. A prominent example is **BABELNET** [99], which provides a mapping of WordNet to a number of collaboratively-constructed resources, including Wikipedia. The structure of BabelNet[3] is similar to that of WordNet. Synsets are the main linguistic units and are connected to other semantically related synsets, whose lexicalizations are multilingual in this case. For instance, the synset corresponding to *United States* is represented with a set of multilingual lexicalizations including $United\_States_{EN}$, $United\_States\_of\_America_{EN}$, $America_{EN}$, $U.S._{EN}$, and $U.S.A._{EN}$ in English, $Estados\_Unidos_{ES}$, $Estados\_Unidos\_de\_América_{ES}$, $EEUU_{ES}$, $E.E.U.U._{ES}$, and $EE.\ UU._{ES}$ in Spanish, and $Stati\_Uniti\_d'America_{IT}$, $Stati\_Uniti_{IT}$, $America_{IT}$, and $U.S.A._{IT}$ in Italian. The relations between synsets are the ones coming from WordNet (hypernyms, hyponyms, etc.), plus new relations coming from other resources such as Wikipedia hyperlinks and WikiData[4] relations (e.g. Madrid *capital of* Spain). BabelNet is the largest multilingual semantic network available, containing 13,789,332 synsets (6,418,418 concepts and 7,370,914 named entities) and 354,538,633 relations for 271 languages[5]. For the English language, BabelNet contains 4,403,148 synsets with at

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

[4] `https://www.wikidata.org`

[5] The statistics are taken from the BabelNet 3.0 release, which is the version used in our experiments. More statistics can be found at `http://babelnet.org/stats`

least one Wikipedia page associated and 117,653 synsets with one WordNet synset associated, from which 99,705 synsets are composed of both a Wikipedia page and a WordNet synset.

The gist of our approach lies in its combination of different types of knowledge from complementary resources. Specifically, our representation approach utilizes the following sources of knowledge: lexico-semantic relations in WordNet, BabelNet's mapping of WordNet synsets and Wikipedia articles, texts within Wikipedia articles and the inter-article links of Wikipedia. In our experiments we used WordNet 3.0 which covers more than 117K unique nouns in about 80K synsets, the Wikipedia dump of December 2014, and BabelNet 3.0, which covers 271 languages and contains over 13 million synsets.

3. Representing texts as vectors

One of the contributions of this article is the framework we are proposing for transforming texts into three different kinds of vector: lexical, embedded and unified. Our lexical vectors follow the conventional approach for representing a linguistic item in a semantic space with words as its dimensions [104] (multiword expressions are also considered). The weights in these vectors are usually computed on the basis of raw term frequencies (*tf*) or normalized frequencies, such as *tf-idf* [52]. Instead, we use lexical specificity for the computation of the weights in our lexical vectors. Having a solid statistical basis, lexical specificity provides several advantages over the previously mentioned measures [17] (see Section 10 for a comparison of lexical specificity and *tf-idf*). In what follows in this section we first explain lexical specificity and propose an efficient way for its fast computation (Section 3.1). We then provide more details of our three types of vector, i.e., lexical (Section 3.2), embedded (Section 3.3) and unified (Section 3.4).

3.1. Lexical specificity

Lexical specificity [62] is a statistical measure based on the hypergeometric distribution[6]. The measure has been widely used in different NLP applications including term extraction [28], textual data analysis [66] and domain-based term disambiguation [14, 10], but it has rarely been used to measure weights in a vector space model. Lexical specificity essentially computes the set of most representative words for a given text based on the hypergeometric distribution. In our setting, we are interested in representing a given text, hereafter referred to as the sub-corpus $\mathcal{SC}$, through a vector comprising the weighted set of its most relevant words or concepts. In order to compute lexical specificity, we need a reference corpus $\mathcal{RC}$ which should be a superset of $\mathcal{SC}$. Lexical specificity computes the weights for each word by contrasting the frequencies of that word across $\mathcal{SC}$ and $\mathcal{RC}$.

Following the notation of [16], let T and t be the respective total number of content words in $\mathcal{RC}$ and $\mathcal{SC}$, while F and f denote the frequency of a given word w in $\mathcal{RC}$ and $\mathcal{SC}$, respectively. Our goal is to compute a weight quantifying the association strength of w with our text $\mathcal{SC}$. We compute the probability of a word w having a frequency equal to or higher than f in our sub-corpus $\mathcal{SC}$ using a hypergeometric distribution which takes as its parameters the frequency of w in the reference corpus $\mathcal{RC}$, i.e., F, and the sizes of $\mathcal{RC}$ and $\mathcal{SC}$, i.e., T and t, respectively. A word w with a high probability is one with a high occurrence chance across arbitrary subsets of $\mathcal{RC}$ of size t. Hence, the representative words of a given sub-corpus will be those with low probabilities since these specific words are the most suitable ones for distinguishing the sub-corpus from the reference corpus. As a result, the computed probability is inversely proportional to the relevance of the word w to $\mathcal{SC}$. In order to make the relation directly proportional, thus making the weights more interpretable, we apply the $-\log_{10}$ operation to the computed probabilities as has been customary in the literature [28, 42]. This logarithmic operation also speeds up the calculations (more details in the following section). Morover, using $\log_{10}$, instead of for instance the natural logarithm, has the added benefit of leading to an easy calculation of the prior probability. For example, if an item has a lexical specificity of 5.0, it means that the probability of observing that item in $\mathcal{SC}$ is $10^{-5} = 0.00005$. Therefore, the lexical specificity of w in $\mathcal{SC}$ is given by the following expression:

$$spec(T, t, F, f) = -\log_{10} P(X \geq f) \tag{1}$$

[6] "The hypergeometric distribution is a discrete probability distribution that describes the probability of k successes in n draws, without replacement, from a finite population of size N that contains exactly K successes, wherein each draw is either a success or a failure. In statistics, the hypergeometric test uses the hypergeometric distribution to calculate the statistical significance of having drawn a specific k successes (out of n total draws) from the aforementioned population" (`https://en.wikipedia.org/wiki/Hypergeometric_distribution`)

where X represents a random variable following a hypergeometric distribution with parameters F, t and T and $P(X \geq f)$ is defined as follows:

$$P(X \geq f) = \sum_{i=f}^{F} P(X = i) \tag{2}$$

where $P(X = i)$ represents the probability of a given word to appear exactly i times in the subcorpus $\mathcal{SC}$ according to the hypergeometric distribution of parameters F, t and T. We propose an efficient implementation of Equation 2 in the following section.

3.1.1. Efficient implementation of lexical specificity

According to Equation 2, the computation of the hypergeometric distribution involves summing $(F - f) + 1$ addends, each of which is calculated as follows[7]:

$$P(X = i) = \frac{\binom{F}{i}\binom{T-F}{t-i}}{\binom{T}{t}} = \frac{F!(T-F)!t!(T-t)!}{T!i!(F-i)!(t-i)!(T-F-t+i)!} \tag{3}$$

Given that the summation range of Equation 2 is generally directly proportional to the size of the corpus, the computation of lexical specificity can be quite expensive on large corpora wherein the value of F tends to be very high. Lafon [62] proposed a method to reduce the computation cost of Equation 2. According to this method, one can first calculate $P(X = i)$ only for the smallest i (i.e., f) and then calculate the rest of probabilities, i.e., $P(X = f + 1)$, ..., $P(X = F)$, using the following property of the hypergeometric distribution:

$$P(X = i + 1) = \frac{P(X = i)(F - i)(t - i)}{(i + 1)(T - F - t + i + 1)} \tag{4}$$

Lafon [62] also suggested using the well-known Stirling formula for the computation of the factorial components in Equation 3. According to the Stirling formula, the logarithm of a factorial can be approximated as follows:

$$\log n! = n \log n - n + \frac{1}{2} \log (2\pi n) \tag{5}$$

Thanks to the application of the Stirling formula we can transform Equation 3 into a summation. Despite these improvements in the calculation of lexical specificity, there remain issues when the above computation is to be applied to a large reference corpus. One of the main problems is the multiplication of potentially very small quantities. Specifically, a 64-bit binary floating-point number, which is the one typically used in current computers, has an approximate range from 10^{-308} through 10^{+308}. During the computation of lexical specificity on large corpora, the lower bound can be reached several times. Our solution to solve this problem (which even optimizes the calculations) is obtained via the next two equations. Firstly, we rewrite Equation 4 by extracting the common factor $P(X = f)$:

$$P(X \geq f) = \sum_{i=f}^{F} P(X = i) = P(X = f) \sum_{i=f}^{F} a_i \tag{6}$$

where $a_f = 1$ and $a_i = a_{i-1} \frac{(F-i)(t-i)}{(i+1)(T-F-t+i+1)}, \forall i = f + 1, ..., F$.

Now we only need to apply the logarithm to both sides of the equation in order to transform the previous multiplication into an addition and thus avoid small values. In this way we also avoid unnecessary exponentials in the calculations of $P(X = f)$:

$$-\log_{10} P(X \geq f) = -\log_{10} P(X = f) - \log_{10} \Big(\sum_{i=f}^{F} a_i \Big) \tag{7}$$

[7] In the cases where $i > t$, which may occur if $F > t$, the probability $P(X = i)$ is equal to 0.

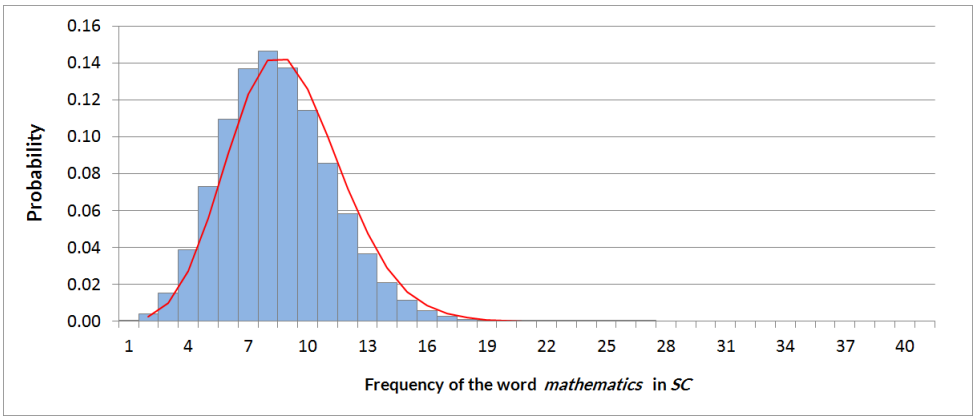

Figure 1: Hypergeometric distribution for the word *mathematics* in an arbitraty sub-corpus ($\mathcal{SC}$) of size 100,000 in Wikipedia.

Therefore, according to Equation 1 and by applying a change of logarithm base, we can compute lexical specificity given the four parameters T, t, F, and f as follows:

$$spec(T, t, F, f) = -k \log_e P(X = f) - \log_{10}\left(\sum_{i=f}^{F} a_i\right) \tag{8}$$

where k is the natural logarithm of 10 (i.e., $\log_e 10$).

For computational feasibility, the $\sum_{i=f}^{F} a_i$ sum is usually not computed until F. Instead, a stopping criterion is introduced into the loop. Since the probability mass in the tail of the hypergeometric distribution is in most cases mathematically insignificant with respect the final cumulative probability distribution, the stopping criterion is usually satisfied well before reaching to the final F value, which considerably reduces the computation time.

As an example we show in Figure 1 the estimated probability distribution for the word *mathematics* in an arbitrary sub-corpus $\mathcal{SC}$ of 100,000 content words from Wikipedia. If the word *mathematics* occurs more than twenty times in $\mathcal{SC}$, the word is considered to be very specific to the given subcorpus, since, as we can see from Figure 1, most of the probability mass in the hypergeometric distribution is concentrated in the left part of the distribution range. The distribution range extends until 70,029, which is the number of occurrences of the word *mathematics* in the whole Wikipedia. However, the probability $P(X = 45)$ is already as small as 10^{-20} and rapidly gets much smaller. This illustrates the point made above, in which the right tail of the probability mass is generally insignificant to values close to the expected value, and adding a stopping condition might make the calculations much faster, while not having any noticeable effect to the final specificity score.

The next three sections provide more details on our three types of vector and on how we leverage lexical specificity for their construction.

3.2. Lexical vector representation

So far we have explained how lexical specificity can be used to determine the relevance of words for a given text. In this section we explain how we leverage lexical specificity in order to construct a lexical vector for a given text (i.e., $\mathcal{SC}$). Throughout the article the texts considered come from Wikipedia, thus we use the whole Wikipedia as our reference corpus ($\mathcal{RC}$). Our lexical vectors have individual words as their dimensions, therefore, in our lexical semantic space, a text is represented on the basis of its association with a set of lexical items, i.e., words. By contrasting the term frequencies across $\mathcal{SC}$ and $\mathcal{RC}$, we compute the lexical specificity of each term for the given subcorpus.

Specifically, in order to compute our lexical vector $\vec{v}_{lex}(\mathcal{SC})$, we simply iterate over all the content words in our subcorpus $\mathcal{SC}$ (only words with a total frequency greater than or equal to five in the whole Wikipedia are considered) and compute lexical specificity for each of them. We then prune the resulting vectors by keeping only those words that are relevant to the target text with a confidence of 99% or more according to the hypergeometric distribution ($P(X \geq f) \leq 0.01$), as also performed in earlier works [10, 17]. Words with weights below the aforementioned threshold are

considered as zero dimensions. The vector truncation step helps reduce noise. Additionally, the truncation helps in speeding up the computation of the vectors, as they will be sparse and therefore computationally easier to work with.

In our setting we also consider multiword expressions when they appear as lexicalizations of piped links[8]. Note that we apply lexical specificity to content words (nouns, verbs and adjectives) after tokenization and lemmatization, but for notational simplicity we will keep using the term "word" to refer to them.

3.3. Embedded vector representation

In recent years, semantic representation has experienced a resurgence of interest in the use of neural network-based learning, a trend usually referred to as word embeddings. In addition to their fast processing of massive amounts of text, word embeddings have proved to be reliable techniques for modeling the semantics of words on the basis of their contexts. However, the application of these word-based techniques to the representation of word senses is not trivial and is bound to the availability of large amounts of sense-annotated data. There have been efforts aimed at learning sense-specific embeddings without needing to resort to sense-annotated data, often through clustering the contexts in which a word appears [139, 47, 100]. However, the resulting representations are usually not aligned to existing sense inventories.

We put forward an approach that allows us to plug in an arbitrary word embedding representation with that of our lexical vector representations, providing three main advantages: (1) benefiting from the word-based knowledge derived as a result of learning from massive corpora for our sense-level representation; (2) reducing the dimensionality of our lexical space to a fixed-size continuous space; and (3) providing a shared semantic space between words and synsets (more details in Section 4), hence enabling a direct comparison of words and synsets.

Our approach exploits the compositionality of word embeddings. According to this property, a compositional phrase representation can be obtained by combining, usually averaging, its constituents' representations [83]. For instance, the vector representation obtained by averaging the vectors of the words *Vietnam* and *capital* is very close to the vector representation of the word *Hanoi* in the semantic space of word embeddings. Our approach builds on this property and plugs a trained word embedding-based representation into our lexical vectors.

Specifically, given an input text $\mathcal{T}$ and a space of word embeddings E, we first calculate the lexical vector of $\mathcal{T}$ (i.e., $\vec{v}_{lex}(\mathcal{T})$) as explained in Section 3.2 and then map our lexical vector to the semantic space E as follows:

$$E(\mathcal{T}) = \frac{\sum_{w \in \vec{v}_{lex}(\mathcal{T})} \left(\frac{1}{rank(w, \vec{v}_{lex}(\mathcal{T}))} E(w) \right)}{\sum_{w \in \vec{v}_{lex}(\mathcal{T})} \frac{1}{rank(w, \vec{v}_{lex}(\mathcal{T}))}} \tag{9}$$

where $E(w)$ is the embedding-based representation of the word w in E, and $rank(w, \vec{v}_{lex}(\mathcal{T}))$ is the rank of the dimension corresponding to the word w in the lexical vector $\vec{v}_{lex}(\mathcal{T})$, thus giving more importance to the higher weighted dimensions. In Section 10 we compare this harmonic average giving more importance to higher weighted words over a simple average. One of the main advantages of this representation combination technique is its flexibility, since any word embedding space can be given as input. As we show in our experiments in Sections 6.1 and 7.1, this combination enables us to benefit from word-specific knowledge and improve it by integrating it into our sense-specific representations.

3.4. Unified vector representation

We also propose a third representation, which we call unified, that, in contrast to the lexical vector representation which has potentially ambiguous words as individual dimensions, has BabelNet synsets as its individual dimensions. Algorithm 1 shows the construction process of a unified vector given the sub-corpus $\mathcal{SC}$. The algorithm first clusters together those words in $\mathcal{SC}$ that have a sense sharing the same hypernym (h in the algorithm) according to the WordNet taxonomy integrated in BabelNet (lines 4-6).

[8] A piped link is a hyperlink which is found within the Wikipedia article that redirects the user to another Wikipedia page. For example, the piped link [[dockside_crane|Crane_(machine)]] is a hyperlink that appears as *dockside_crane* in the text, but links to the Wikipedia page titled *Crane_(machine)*. The Wikipedia article is therefore represented with a suitable lexicalization that preserves the grammatical and syntactic structure, the contextual coherency and the flow of the sentence.

Algorithm 1 Unified Vector Construction

Input: A reference corpus $\mathcal{RC}$ and a sub-corpus $\mathcal{SC}$
Output: the unified vector $\vec{u}_s$ where $\vec{u}_s(h)$ is the dimension corresponding to the synset h

1: $T \leftarrow size(\mathcal{RC})$
2: $t \leftarrow size(\mathcal{SC})$
3: $H \leftarrow \emptyset$
4: **for each** lemma $l \in \mathcal{SC}$
5: **for each** hypernym h of l in BabelNet
6: $H \leftarrow H \cup \{h\}$
7: $\vec{u} \leftarrow$ empty vector
8: **for each** $h \in H$
9: **if** $\exists\, l_1, l_2 \in \mathcal{SC}$: l_1, l_2 hyponyms of h and $l_1 \neq l_2$ **then**
10: $F \leftarrow 0$
11: $f \leftarrow 0$
12: $hyper_{pass} \leftarrow False$
13: **for each** lexicalization lex of h
14: $F \leftarrow F + freq(lex, \mathcal{RC})$
15: $f \leftarrow f + freq(lex, \mathcal{SC})$
16: $spec_h \leftarrow specificity(T, t, freq(lex, \mathcal{RC}), freq(lex, \mathcal{SC}))$
17: **if** $spec_h \geq spec_{thres}$ **then**
18: $hyper_{pass} \leftarrow True$
19: **if** $hyper_{pass}$ **then**
20: **for each** hyponym $hypo$ of h
21: **for each** lexicalization lex of $hypo$
22: $F \leftarrow F + freq(lex, \mathcal{RC})$
23: $f \leftarrow f + freq(lex, \mathcal{SC})$
24: $\vec{u}(h) \leftarrow specificity(T, t, F, f)$
25: **return** $\vec{u}$

On all hyponym clusters we impose the restriction that they should have at least one lexicalization of the hypernym above the standard lexical specificity threshold 2 (lines 16-18). The reason why we include this in the unified representation is to reduce some noise detected by applying the old unified algorithm [16]. Finally, if the cluster passes the threshold, the specificity is computed for the set of all the hyponyms of h, even those which do not occur in the sub-corpus $\mathcal{SC}$ (lines 20-24). As in Section 3.1, F and f denote the frequencies in the reference corpus $\mathcal{RC}$ (Wikipedia) and the sub-corpus $\mathcal{SC}$, respectively. In this case, the frequencies correspond to the aggregation of frequencies of h and all its hyponyms.

Our clustering of sibling words into a single cluster represented by their common hypernym transforms a lexical space into a unified semantic space. This space has multilingual synsets as dimensions, enabling their direct comparability across languages. We evaluated this feature of the unified vectors on the task of cross-lingual word similarity in Section 6.1.3. The clustering may also be viewed as an implicit disambiguation of potentially ambiguous words, as they are disambiguated into their intended sense represented by their hypernym, resulting in a more accurate semantic representation.

3.5. Vector comparison

As our vector comparison method for the lexical and unified vectors we use the square-rooted Absolute Weighted Overlap [17, 16], which is based on the Weighted Overlap measure [107]. For notational brevity, we will refer to the square-rooted Absolute Weighted Overlap as Weighted Overlap (WO). WO compares two vectors on the basis of their overlapping dimensions, which are harmonically weighted by their absolute rankings. For this measure the vectors are viewed as *semantic sets* or *ranked lists* [136], as the weights are only used to sort the elements within the vector and their actual values are not used in the calculation. Formally, Weighted Overlap between two vectors $\vec{v_1}$ and $\vec{v_2}$ is defined as follows:

$$WO(\vec{v_1}, \vec{v_2}) = \sqrt{\frac{\sum_{d \in O} \left(rank(d, \vec{v_1}) + rank(d, \vec{v_2})\right)^{-1}}{\sum_{i=1}^{|O|} (2i)^{-1}}} \tag{10}$$

where O is the set of overlapping dimensions (i.e., concepts or words) between the two vectors and $rank(d, \vec{v_i})$ is the rank of dimension d in the vector $\vec{v_i}$. Absolute WO differs from the original WO, which takes into account the relative ranks of the dimensions with respect to the overlapping dimensions, instead of considering all the dimensions of the vector. Owing to the use of absolute ranks this measure gives lower scores in comparison to the original WO. This is the reason behind the use of the square-root operator, which smooths the distribution of values over the [0,1] scale. This metric has been shown to suit specificity-based vectors more than the conventional cosine distance [17].

In contrast, we use cosine for comparing our embedded vector representations. The dimensions of the embedded representations are not interpretable and the dimension values do not represent weights, thus rank-based WO is not applicable on this setting. Cosine is the usual measure used in the literature to measure similarity in an embedding space [82, 19, 68].

4. From a synset to its vector representations

In Section 3 we proposed three vector representations of an arbitrary text or subcorpus $\mathcal{SC}$ belonging to a larger collection. We now see how we leverage these representations to obtain a semantic vector representation for concepts and named entities. As knowledge base we use BabelNet[9], a multilingual encyclopedic dictionary which merges WordNet with other lexical and encyclopedic resources such as Wikipedia and Wiktionary, thanks to its use of an automatic mapping algorithm [98, 99]. We chose BabelNet due to its large coverage of named entities and concepts in hundreds of languages. Moreover, concepts and named entities are organized into a full-fledged taxonomy which integrates the WordNet taxonomy, which is the one used in our experiments, and, from its latest versions, the Wikipedia Bitaxonomy [34], WikiData, and *is-a* relations coming from open information extraction techniques [26]. Our approach makes use of the full power of BabelNet, as it exploits the complementary information of the distributional statistics in Wikipedia articles that are tied to the taxonomical relations in BabelNet. In our experiments, we used version 3.0 of BabelNet (released in December 2014) which covers around 6.5M concepts and more than 7M named entities in 271 different languages. The rest of this section is divided into two parts. We first show how we collect contextual information for a given synset (Section 4.1) and then explain how this contextual information is processed in order to obtain our vector representations (Section 4.2).

4.1. Getting contextual information for a given synset

The goal of the first step is to create a subcorpus $\mathcal{SC}_s$ for a given BabelNet synset s. Let $\mathcal{W}_s$ be the set containing the Wikipedia page corresponding to the concept s (wp_s henceforth) and all the related Wikipedia pages that have an outgoing link to that page. Note that at this stage $\mathcal{W}_s$ might be empty if there is no Wikipedia page corresponding to the BabelNet synset s. We further enrich $\mathcal{W}_s$ by adding the corresponding Wikipedia pages of the hypernyms and hyponyms of s in the taxonomy of BabelNet. Figure 2 illustrates our procedure for obtaining contextual information. Let $\mathcal{SC}_s$ be the set of content words occurring in the Wikipedia pages of $\mathcal{W}_s$ after tokenization and lemmatization. The frequency of each content word w of $\mathcal{SC}_s$ is calculated as follows:

$$f(w) = \sum_{i=1}^{n} \lambda_i f_i(w) \tag{11}$$

where n is the number of Wikipedia pages in $\mathcal{W}_s$, $f_i(w)$ is the frequency of w in the Wikipedia page $p_i \in \mathcal{W}_s$ (i=1,...,n), and λ_i is the weight assigned to the page p_i to denote its importance. In the following subsection we explain how we calculate the weight λ_i for a given page p_i.

4.1.1. Weighting semantic relations

In this section we explain how we weight the BabelNet semantic relations (i.e., λ_i in Equation 11) between the target synset s and the i-th page in $\mathcal{W}_s$. In previous versions of Nasari [17, 16] we were making an assumption that all the Wikipedia pages in $\mathcal{W}_s$ were equally important (i.e., $\lambda_i = 1, \forall i \leq n$). In this article we set more meaningful weights for these pages on the basis of the source and type of semantic connection to the target synset s.

[9] See Section 2 for more information about BabelNet.

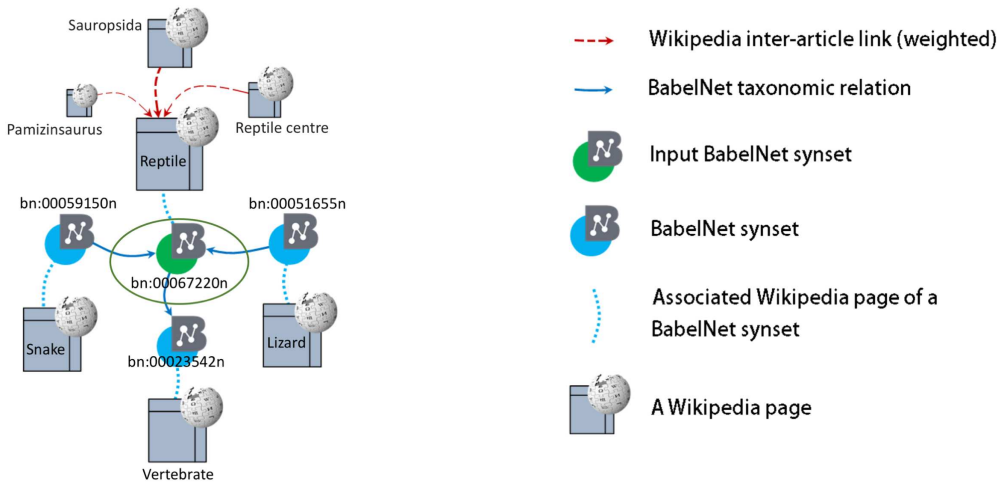

Figure 2: Our procedure for getting contextual information of the sample BabelNet synset represented by its main sense $reptile_n^1$.

A Wikipedia page in $\mathcal{W}_s$ may come from three different sources (see Section 4.1): (1) the Wikipedia page corresponding to s (wp_s), (2) the related Wikipedia pages that have an outgoing link to the page wp_s, and (3) the Wikipedia pages that are connected to s through taxonomic relations in BabelNet, We compute and assign a weight in the [0, 1] range for the pages of each type as follows:

1. The **Wikipedia page corresponding to the BabelNet synset** s (i.e., wp_s) is assigned the highest possible weight of 1.
2. The weights for the **related Wikipedia pages that have an outgoing link to** wp_s are computed as follows. We first compute the lexical vectors of these Wikipedia pages, as well as for wp_s. We then apply Weighted Overlap (see Section 3.5) to calculate the similarity between the lexical vectors of each of these pages and that of wp_s. These similarity scores denote the weight of each related Wikipedia page. In order to reduce the high number of ingoing links in some cases, and to improve the quality of these links, we prune the ingoing links to include only the top 100 links on the basis of their similarity scores and those whose similarity score is higher than 0.25.
3. Given there is a possibility that a particular synset does not have a Wikipedia page associated with it, the Wikipedia pages coming from taxonomic relations cannot be calculated as in the previous case. In this case, the **Wikipedia pages coming from taxonomic relations** are given a fixed score of 0.85, which was calculated as follows. We picked a set of 100 random taxonomic relations and calculated the average similarity score among the 100 pairs by using our previous NASARI system.

4.2. Transforming the contextual information into vector representations

Once we have gathered a corpus $\mathcal{SC}_s$ for a given BabelNet synset s and computed the associated frequencies $f(w)$ for each word w in $\mathcal{SC}_s$, we proceed to calculate the lexical, embedded and unified vectors of s as explained in Sections 3.2, 3.3 and 3.4, respectively. In our experiments, we used the whole Wikipedia corpus as our reference corpus $\mathcal{RC}$ (Wikipedia dump of December 2014)[10]. We computed NASARI lexical and unified vectors for English, German, French, Italian, and Spanish. The number of synset vectors for each of these languages is, respectively, 4.42M, 1.51M, 1.48M, 1.10M and 1.07M. On average, for the English language, the contextual information of a synset is composed of a subcorpus $\mathcal{SC}_s$ of 1561 words in total coming from 17 Wikipedia pages. For the embedded vectors, we took as word embeddings the pre-trained word and phrase vectors from Word2Vec[11]. These vectors were trained on a 100-billion English corpus from Google News and have 300 dimensions.

[10] Each language uses the Wikipedia corpus in its respective language as reference corpus.
[11] The pre-trained Word2Vec word embeddings were downloaded at `https://code.google.com/p/word2vec/`.

Bank (financial institution)			**Bank (geography)**		
English	French	Spanish	English	French	Spanish
bank	banque	banco	river	eau	banco
banking	bancaire	bancario	stream	castor	limnología
deposit	crédit	banca	bank	berge	ecología
credit	financier	financiero	riparian	canal	barrera
money	postal	préstamo	creek	barrage	estuarios
loan	client	entidad	flow	zone	isla
commercial_bank	dépôt	déposito	water	perchlorate	interés
central_bank	billet	crédito	watershed	humide	laguna

Table 1: Top-weighted dimensions from the lexical vectors of the financial and geographical senses of *bank*.

Bank (financial institution)			**Bank (geography)**		
English	French	Spanish	English	French	Spanish
‡bank$^{2}_{n}$	‡banque$^{1}_{n}$	‡banco$^{1}_{n}$	⋆stream$^{1}_{n}$	eau$^{1}_{n}$	inclinación$^{9}_{n}$
reserve$^{2}_{n}$	•fonds$^{2}_{n}$	⋆Institución_financiera$^{1}_{n}$	river$^{1}_{n}$	eau$^{15}_{n}$	lago$^{1}_{n}$
⋆financial_institution$^{1}_{n}$	⋄dépôt$^{9}_{n}$	⋄depósito$^{15}_{n}$	‡body_of_water$^{1}_{n}$	excrément$^{1}_{n}$	‡cuerpo_de_agua$^{1}_{n}$
⋄deposit$^{8}_{n}$	∘emprunt$^{2}_{n}$	†Finanzas$^{1}_{n}$	flow$^{1}_{n}$	castor$^{1}_{n}$	⋆arroyo$^{1}_{n}$
banking$^{2}_{n}$	paiement$^{1}_{n}$	•dinero$^{2}_{n}$	course$^{2}_{n}$	‡étendue_d'eau$^{1}_{n}$	tierra$^{11}_{n}$
†finance$^{1}_{n}$	argent$^{2}_{n}$	∘préstamo$^{2}_{n}$	bank$^{1}_{n}$	fourrure$^{1}_{n}$	costa$^{1}_{n}$

Table 2: Top-weighted dimensions from the unified vectors of the financial and geographical senses of *bank*. We represent each synset by one of its word senses. Word senses marked with the same symbol across languages correspond to the same BabelNet synset.

Lexical and unified synset vectors example. We show in Tables 1 and 2, respectively, the top-weighted dimensions of the lexical and unified vector representations for the financial and geographical senses of the noun *bank* in three different languages, i.e., English, French and Spanish. As can be seen, the two senses of *bank* are clearly identified and distinguished from each other according to the top dimensions of their vectors, irrespective of their language and type. Additionally, note that the unified vectors are comparable across languages. We mark in Table 2, across different languages, those word senses[12] that correspond to the same BabelNet synset. It can be seen from the Table that the unified vectors in different languages share many of their top elements.

Word and synset embeddings example. The dimensions are not interpretable in the embedded vectors. Therefore, a better way to distinguish different senses would be to show their closest elements in the space (using cosine as vector similarity measure). Table 3 shows the eight closest senses to the word *bank*, as well as those closest to two specific senses of this word, i.e., the financial and geographical senses (recall that in our embedded vector representation words and synsets share the same space). In this case, both senses of *bank* are again clearly distinguished by their closest BabelNet synsets in the space. Looking at the closest senses to the word *bank* we can see that most of these are rather somehow to the financial meaning of *bank*, with lower cosine values, though. This shows that the predominant sense of the word *bank* in the Google News corpus (on which the word embeddings are trained) is clearly its financial sense. We note that using our embedded vector representation one can easily compute the predominance of the senses of a word by directly comparing the representation of that word with those of its individual senses. Our shared space also provides a suitable framework for studying the ambiguity of words.

5. Summary of the Experiments

In order to assess the reliability and flexibility of our technique across different datasets and tasks, we carried out a comprehensive set of evaluations. Specifically, we considered four different tasks: Semantic Similarity (Section

[12] We use the sense notation of [94]: $word^{p}_{n}$ is the n^{th} sense of the *word* with part of speech p.

Bank (financial institution)		**Bank (geography)**		***bank***	
Closest senses	Cosine	Closest senses	Cosine	Closest senses	Cosine
Deposit account	0.99	Stream bed	0.98	Bank (financial institution)	0.86
Universal bank	0.99	Current (stream)	0.97	Universal bank	0.86
British banking	0.98	River engineering	0.97	British banking	0.86
German banking	0.98	Braided river	0.97	German banking	0.85
Commercial bank	0.98	Fluvial terrace	0.97	Branch (banking)	0.85
Banking in Israel	0.98	Bar (river morphology)	0.97	McFadden Act	0.85
Financial institution	0.98	River	0.97	Four Northern Banks	0.84
Community bank	0.97	Perennial stream	0.96	State bank	0.84

Table 3: Closest embedded vectors from the BabelNet synsets corresponding to the financial and geographical senses of *bank*, and from the word *bank*.

6), Sense Clustering (Section 7), Domain Labeling (Section 8) and Word Sense Disambiguation (Section 9). A brief overview of the evaluation benchmarks and the results across the four tasks follows:

1. **Semantic similarity.** NASARI proved to be highly reliable in the task of semantic similarity measurement, as it provides state-of-the-art performance on several datasets across different evaluation benchmarks:

   - *Mono-lingual word similarity* on four standard word similarity datasets, namely, MC-30 [85], WS-Sim [33], SimLex-999 [43] and RG-65 [118]. In addition to these English word similarity datasets, we also assessed the multilinguality of our approach on the RG-65 dataset in three other languages.
   - *Cross-lingual word similarity* on six different cross-lingual datasets on the basis of RG-65 [15].

   In addition to the above three word similarity benchmarks, we also assessed the capability of our approach to provide comparable semantic representations for different types of linguistic items. Specifically, we opted for the SemEval-2014 task on *Cross-Level Semantic Similarity* [57]. Despite not being tuned for this task, our approach achieved near state-of-the-art performance on the word to sense similarity measurement dataset.

2. **Sense clustering.** We constructed a highly competitive unsupervised system on the basis of the NASARI representations, outperforming state-of-the-art supervised systems on two manually-annotated Wikipedia sense clustering datasets [23].

3. **Domain labeling.** We used our system for annotating synsets of a large lexical semantic resource (BabelNet), and benchmarked our system against three automatic baselines on two gold standard datasets: a dataset of domain-labeled WordNet synsets coming from WordNet 3.0, and a new manually-constructed dataset of domain-labeled BabelNet synsets. NASARI outperformed all automatic baselines, demonstrating that our approach is not only reliable, but is also flexible across different tasks.

4. **Word Sense Disambiguation.** We proposed a simple framework for a knowledge-rich unsupervised disambiguation system. Our system obtained state-of-the-art results on multilingual All-Words Word Sense Disambiguation using Wikipedia as sense inventory, evaluated on the SemEval-2013 dataset [96], and on English All-Words Word Sense Disambiguation using WordNet as sense inventory, evaluated on the SemEval-2007 [112] and SemEval-2013 [96] datasets. Additionally, we performed an experiment to measure the reliability of our semantic representations for named entities, obtaining the best results among all unsupervised systems and near state-of-the-art performance on the SemEval-2015 WSD dataset [89].

6. Semantic Similarity

Semantic similarity is the most popular benchmark for the evaluation of different semantic representation techniques. The task here is to measure the semantic closeness of two linguistic items. The similarity of two items can

be directly computed by comparing their corresponding vector representations. As we mentioned in Section 3.5, we opted for Weighted Overlap as our vector comparison method for lexical and unified representations, and cosine for the embedded representations. Note that by using our approach we obtain representations for individual BabelNet synsets. Moreover, because BabelNet merges different resources, our representations can be used to calculate the semantic similarity between any two semantic units within and across different resources, for instance between two Wikipedia pages, two WordNet synsets, or a Wikipedia page and a WordNet synset.

6.1. Evaluation

We benchmark our semantic similarity procedure on the word similarity task. Word similarity is a specific task from semantic similarity in which we measure how semantically close two words are. In order to be able to compute the similarity between words we first need to map the two words to their corresponding synsets. However, this mapping is a straightforward process, thanks to the multilingual sense inventory of BabelNet. As frequently done in this task, we measure the similarity between two words w and w' as the similarity between their closest senses [116, 13, 107, 17]:

$$sim(w, w') = \max_{\vec{v_1} \in \mathcal{L}_w, \vec{v_2} \in \mathcal{L}_{w'}} VC(\vec{v_1}, \vec{v_2}) \tag{12}$$

where $\mathcal{L}_w$ represents the set of synsets which contain w as one of its lexicalizations. As vector comparison VC we use WO (see Section 3.5) to compare lexical and unified representations, and cosine for the embedded representations.

Note that, thanks to our unified representation, w and w' may belong to different languages. Throughout this section on the tasks based on semantic similarity, **NASARI$_{\textbf{lexical}}$** and **NASARI$_{\textbf{unified}}$** represent the systems based on the lexical and unified vectors, respectively. We refer to the combination of both lexical and unified vectors as **NASARI**. This combination is based on the average similarity scores given by lexical and unified vectors for each sense pair. We also report results of our **NASARI$_{\textbf{embed}}$** vector representations which use the pre-trained Word2Vec vectors as input. We performed experiments on monolingual word similarity for English and other languages (presented in Sections 6.1.1 and 6.1.2, respectively) and cross-lingual similarity (presented in Section 6.1.3). Additionally, we evaluate our embedded representations in a cross-level semantic similarity task in Section 6.1.4.

6.1.1. Monolingual word similarity: English

Datasets. The majority of benchmarks for word similarity are available only for the English language. We compare our approach with other state-of-the-art word similarity systems on standard English word similarity datasets. We chose the standard MC-30 [85], WordSim-353 [33], and SimLex-999 [43] as evaluation benchmarks. **MC-30** consists of a subset of RG-65 [118] which was re-annotated following new similarity guidelines. WordSim-353 consists of 353 word pairs, including both concepts and named entities. In the original WordSim-353 similarity conflated relatedness in the same dataset. In order to avoid this conflation, [1] cleverly divided the dataset into two subsets: the first one concerned relatedness while the second subset focused on similarity, the latter being the one used in our experiments. We will refer to this similarity subset of 203 word pairs as **WS-Sim** henceforth. Finally, we took the noun pairs from the **SimLex-999** dataset as our last evaluation benchmark. The complete SimLex-999 dataset is composed of 999 word pairs, 666 of which are noun pairs.

Comparison systems. We selected state-of-the-art approaches which are available online as comparison systems. These systems can be split into two categories: knowledge-based and corpus-based. As knowledge-based, we selected two approaches based on the WordNet semantic graph: [107, **ADW**][13] and [69, **Lin**][14]. Another knowledge-based approach is [35, **ESA**][15], which represents a word in a semantic space of Wikipedia articles. We also compared our systems with four corpus-based approaches[16]. Firstly, we took the pre-trained *word embeddings* of **Word2Vec**[82][17], the same used for our NASARI$_{\text{embed}}$ system (see Section 4.2). Then, we took the best predictive and count-based

[13] ADW implementation available at `https://github.com/pilehvar/ADW`

[14] Results for Lin were obtained from the WS4J implementation available at `https://code.google.com/p/ws4j/`

[15] ESA implementation available at DKProSimilarity package [7].

[16] All the corpus-based approaches mentioned in the paper use cosine as comparison measure.

[17] The pre-trained models are available at `https://code.google.com/p/word2vec/`. They were trained on a Google News corpus of about 100 billion words.

	MC-30		WS-Sim		SimLex-999 (nouns)		Average	
	r	ρ	r	ρ	r	ρ	r	ρ
NASARI	0.89	0.78	0.74	0.72	0.50	0.49	**0.71**	0.67
$\text{NASARI}_{\text{lexical}}$	0.88	0.81	0.74	0.73	0.51	0.49	**0.71**	**0.68**
$\text{NASARI}_{\text{unified}}$	0.88	0.78	0.72	0.70	0.49	0.48	0.70	0.65
$\text{NASARI}_{\text{embed}}$	**0.91**	0.83	0.68	0.68	0.48	0.46	0.69	0.66
ESA	0.59	0.65	0.45	0.53	0.16	0.23	0.40	0.47
Lin	0.76	0.72	0.66	0.62	**0.58**	**0.58**	0.67	0.64
ADW	0.79	0.83	0.63	0.67	0.44	0.45	0.62	0.65
Chen	0.82	0.82	0.63	0.64	0.48	0.44	0.64	0.63
Word2Vec	0.80	0.80	**0.76**	**0.77**	0.46	0.45	0.67	0.67
Best-Word2Vec	$0.83^{\ddagger}$	$0.83^{\ddagger}$	$0.76^{\ddagger}$	$0.78^{\ddagger}$	0.48	0.49	0.69	0.70
Best-PMI-SVD	$0.76^{\ddagger}$	$0.71^{\ddagger}$	$0.68^{\ddagger}$	$0.66^{\ddagger}$	0.40	0.40	0.61	0.59
SensEmbed	0.89	**0.88**	0.65	0.75	$0.46^{\dagger}$	$0.47^{\dagger}$	0.67	0.70
IAA	-	-	-	$0.61^{\diamond}$	-	0.61		

Table 4: Pearson (r) and Spearman (ρ) correlations of different similarity measures with human judgements on RG-65, MC-30, WS-Sim and SimLex-999 (noun instances) datasets. We show the best performance obtained by [8] out of 48 configurations across different datasets including WS-Sim and RG-65 (highlighted by ‡). We show the SenseEmbed configuration tuned on the SimLex-999 dataset (highlighted by †). The inter-annotator agreement of the whole WordSim-353 (highlighted with $\diamond$) was reported to be 0.61, no inter-annotator agreement has been reported for the WS-Sim subset.

models for semantic similarity released by [8][18]. The best predictive model is based on Word2Vec (**Best-Word2Vec** henceforth), while the best count-based models (**PMI-SVD**) are traditional co-occurrence vectors based on Pointwise Mutual Information (PMI) combined with a Singular Value Decomposition (SVD) dimensionality reduction. Finally, we benchmarked our system against two embedding-based sense representation approaches. The first approach, **Chen** henceforth [19], leverages word embeddings, WordNet glosses and a WSD system for creating sense embeddings[19]. The second one, called **SensEmbed** [48], uses BabelNet as the main knowledge source and also relies on pre-disambiguated text by using a WSD system. We report the results of these last two methods when using the same closest senses strategy used by our systems.

Results. Table 4 shows Pearson and Spearman correlation performance of our systems and all comparison systems on the three considered datasets[20]. Both lexical and unified vectors, especially the lexical ones, prove to be quite robust across datasets. The combination of both lexical and unified vectors does not show any noticeable improvement over the lexical vectors single-handed. Our system gets the highest average Pearson correlation among all systems, outperforming even the embedding-based approaches which use one dataset (SensEmbed) or two datasets (Best-Word2Vec) in order to tune their hyperparameters[21]. In terms of Spearman correlation, our system based on the lexical vectors also achieves the highest average performance among the systems which do not use any of the datasets for tuning with a single point advantage over Word2Vec. $\text{NASARI}_{\text{embed}}$ also proves to be quite competitive, outperforming all Word2Vec approaches in terms of Pearson correlation and obtaining the best overall result on MC-30.

Lin, which does not perform particularly well on MC-30 and WS-Sim, surprisingly obtains the best overall performance on the SimLex-999 dataset, which is largest considered dataset, consisting of 666 noun pairs. Our system gets the second best overall performance on this dataset. A closer look at the output of the similarity scores given by

[18]Both models were trained on a 2.8 billion-token corpus including the English Wikipedia. They are available at `clic.cimec.unitn.it/composes/semantic-vectors.html`

[19]The sense representations were downloaded from `http://pan.baidu.com/s/1eQcPK8i`

[20]Inter-annotator agreement (IAA) is also reported for those datasets for which this information is available. IAA is reported in terms of average pairwise Spearman correlation.

[21][67] showed that with a fine tuning, Word2Vec can achieve a 0.79 Spearman correlation performance on WS-Sim, higher than the 0.77 Spearman correlation reported by [8] on that dataset.

English	r	ρ	**French**	r	ρ	**German**	r	ρ	**Spanish**	r	ρ
NASARI	0.81	0.78	NASARI	**0.82**	0.73	NASARI	0.69	0.65	NASARI	**0.85**	0.79
NASARI$_{\text{lexical}}$	0.80	0.78	NASARI$_{\text{lexical}}$	0.80	0.70	NASARI$_{\text{lexical}}$	0.69	0.67	NASARI$_{\text{lexical}}$	**0.85**	0.79
NASARI$_{\text{unified}}$	0.80	0.76	NASARI$_{\text{unified}}$	**0.82**	**0.76**	NASARI$_{\text{unified}}$	**0.71**	**0.68**	NASARI$_{\text{unified}}$	0.82	0.77
NASARI$_{\text{embed}}$	**0.82**	**0.80**	–	–	–	–	–	–	NASARI$_{\text{embed}}$	0.79	0.77
SOC-PMI	0.61	–	SOC-PMI	0.19	–	SOC-PMI	0.27	–	–	–	–
PMI	0.41	–	PMI	0.34	–	PMI	0.40	–	–	–	–
LSA-Wiki	0.65	0.69	LSA-Wiki	0.57	0.52	–	–	–	–	–	–
Wiki-wup	0.59	–	–	–	–	Wiki-wup	0.65	–	–	–	–
Word2Vec	–	0.73	Word2Vec	–	0.47	Word2Vec	–	0.53	Best-Word2Vec	0.80	**0.80**
Retrofitting	–	0.77	Retrofitting	–	0.61	Retrofitting	–	0.60	–	–	–
NASARI$_{\text{poly-embed}}$	0.74	0.77	NASARI$_{\text{poly-embed}}$	0.60	0.69	NASARI$_{\text{poly-embed}}$	0.46	0.52	NASARI$_{\text{poly-embed}}$	0.68	0.74
Polyglot-embed	0.51	0.55	Polyglot-embed	0.38	0.35	Polyglot-embed	0.18	0.15	Polyglot-embed	0.51	0.56
IAA	0.85$^{\diamond}$	-	IAA	-	-	IAA	0.81	-	IAA	0.83	-

Table 5: Pearson (r) and Spearman (ρ) correlation performance of different systems on the English, French, German and Spanish RG-65 datasets. The inter-annotator of the English RG-65 (highlighted with $\diamond$) was calculated for a subset of fifteen annotators.

our system compared to the gold standard shows noticeable errors when measuring the similarity between antonym pairs, which are heavily represented in this dataset. These antonym pairs were given consistently low values across the dataset, irrespective of the target words, whereas we argue that the similarity scores ought to vary according to the particular semantics of the antonym pairs. For instance, the pair *day-night* gets a score of 1.9 in the 0-10 scale, while our system gets a much higher 8.0 score[22]. A similar phenomenon is found on the *sunset-sunrise* pair. Nevertheless, in both cases the words in the pair belong to coordinate synsets in WordNet. In fact, recent works [122, 106, 92] have shown how significant performance improvements can be obtained on this dataset by simply tweaking usual word embedding approaches to handle antonymy. This differs from the scores given in the WordSim-353 dataset, in which antonym pairs were considered as similar [43]. It is outside the scope of this work to change this feature of our system in order to resolve its judgment differences with respect to the human annotation of antonym pairs in the SimLex-999 dataset.

6.1.2. Multilingual word similarity

Datasets. We took the **RG-65** dataset as evaluation benchmark. The language of this dataset was originally English [118]. It was later translated into French [54], German [39] and Spanish [15]. We used the four versions of the dataset for our experiments.

Comparison systems. We benchmark our system against other multilingual word similarity approaches. **Wiki-wup** [111] and **LSA-Wiki** [38] are systems which use Wikipedia as their main knowledge resource. We also provide results for co-occurrence-based methods such as **PMI** and **SOC-PMI** [54] and for the newer word embeddings [31]. For word embeddings we report results for the **Word2Vec** model[23] and for an approach retrofitting these Word2Vec vectors into WordNet (**Retrofitting**) [31]. For the Spanish language no result was reported in [31] for Word2Vec, so we trained Word2Vec with the same hyperparameters of **Best-Word2Vec** [8] on the *Spanish Billion Words Corpus*[24] [18]. We used these Spanish word embeddings as input for our NASARI$_{\text{embed}}$ system in this language. Additionally, we report results for pre-trained embeddings in all four languages [5, **Polyglot-embed**][25]. These vectors have sixty-four dimensions and were trained on the Wikipedia corpus. We also compare this system with our embedded representations of synsets by using the *polyglot* word embeddings as input continuous representations (see Section 3.3). We will refer to this latter method as NASARI$_{\text{poly-embed}}$.

[22] All scores have been converted to the 0-10 scale for this example.

[23] For English, the pre-trained models of Word2Vec trained on a Google News corpus of 100 billion words were considered for the evaluation. For French and German, a corpus of a 1 billion tokens from Wikipedia was used for training.

[24] Downloaded from `http://crscardellino.me/SBWCE/`

[25] The pre-trained polyglot word representations were downloaded from `https://sites.google.com/site/rmyeid/projects/polyglot`.

Measure	EN-FR		EN-DE		EN-ES		FR-DE		FR-ES		DE-ES		Average	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
NASARI$_{\text{unified}}$	**0.84**	0.79	**0.79**	0.79	**0.84**	0.82	0.75	0.70	**0.86**	0.78	**0.81**	**0.80**	**0.82**	0.78
CL-MSR-2.0	0.30	–	–	–	–	–	–	–	–	–	–	–	–	–
NASARI$_{pivot}$	0.79	0.69	0.78	0.76	0.80	0.74	**0.79**	0.70	0.80	0.67	0.72	0.68	0.78	0.71
ADW$_{pivot}$	0.80	0.82	0.73	**0.82**	0.78	**0.84**	0.72	**0.77**	0.81	**0.81**	0.68	0.72	0.75	**0.80**
Word2Vec$_{pivot}$	0.77	0.82	0.70	0.73	0.76	0.80	0.65	0.70	0.75	0.76	0.64	0.63	0.71	0.74
Best-Word2Vec$_{pivot}$	0.75	**0.84**	0.69	0.76	0.75	0.82	0.77	0.73	0.74	0.79	0.64	0.64	0.72	0.76
Best-PMI-SVD$_{pivot}$	0.76	0.76	0.72	0.74	0.77	0.77	0.65	0.69	0.76	0.74	0.62	0.61	0.71	0.72

Table 6: Pearson (r) and Spearman (ρ) correlation performances of different similarity measures on the six cross-lingual RG-65 datasets. Notation: English (EN), French (FR), German (DE), Spanish (ES).

Results. Table 5 shows Pearson and Spearman correlation performance of our systems and all comparison systems on the RG-65 word similarity datasets for English, French, German and Spanish[26]. Our system outperforms all multilingual comparison systems in English, French and German in terms of both Pearson and Spearman correlation. For the Spanish language our system surprisingly slightly outperforms the human inter-annotator agreement (which was calculated in terms of average pairwise Pearson correlation), hence demonstrating the competitiveness of our approach in this language too.

The Polyglot-embed multilingual representations do not show a particular potential for the task. The reason behind these results may be due, apart from the inherent ambiguity of words, to their low dimensionality (64) and small vocabulary (100K words). However, our embedded representation using these word embeddings (NASARI$_{\text{poly-embed}}$) hugely improves the original vectors (obtaining an average twenty-three Pearson and twenty-eight Spearman correlation points improvement). NASARI$_{\text{poly-embed}}$, despite achieving lower results than our three representations, achieves competitive results with respect to other comparison systems, with the added benefit of being applicable to many languages (pre-trained polyglot embeddings are available for more than a hundred languages).

6.1.3. Cross-lingual word similarity

Datasets. We have chosen the RG-65 cross-lingual datasets released by [15] for English, French, German and Spanish. These datasets[27] were automatically constructed by taking the manually-curated multilingual RG-65 datasets from the previous Section as input. In total, we evaluated on six datasets consisting of all the possible language pair combinations for the four languages.

Comparison systems. As cross-lingual comparison systems, we have included the best results provided by the **CL-MSR-2.0** system [60]. This system applies PMI on an English-French parallel corpus obtained from WordNet. Additionally, we provide results for some of the best performing systems in English word similarity by using English as a pivot language[28]. Baseline pivot systems include the WordNet-based system **ADW** [107], the pre-trained **Word2Vec** word embeddings [82] and the top performing Word2Vec model in similarity obtained by [8] (**Best-Word2Vec**), and the best count-based model obtained by [8] (**PMI-SVD**). See Section 6.1.1 for more details on these comparison systems. We also report results for our system using the combination of lexical and unified English NASARI vectors. We refer to all these systems using English as pivot language as *pivot*.

Results. Table 6 shows cross-lingual word similarity results according to Pearson and Spearman correlation performance. In this section we only report results for our unified vector representations, as their dimensions are BabelNet synsets, which are multilingual and therefore may be used for direct cross-lingual comparison. Our unified vector representations outperform all comparison systems (both types) in terms of Pearson correlation performance except

[26]Inter-annotator agreement (IAA) is also reported for the languages for which this information is available. IAA is reported in terms of average pairwise Pearson correlation.

[27]The cross-lingual datasets are available at `http://lcl.uniroma1.it/similarity-datasets/`

[28]Non-English words are translated by using Google Translate.

for the French-German pair, in which our *pivot* system obtains the best result. It is interesting to note that our English monolingual similarity proves to be the most robust across language pairs among all *pivot* systems according to Pearson correlation measure, demonstrating the reliability of our system also on a purely monolingual scheme. *Pivot* systems prove to be competitive, outperforming the only cross-lingual baseline which does not use a pivot language. In fact, despite obtaining relatively modest Pearson results, ADW obtains the best results according to the Spearman correlation measure (our unified vector representations obtain the second best result overall). In terms of the harmonic mean of Pearson and Spearman, used as official measure in a previous semantic similarity SemEval task [57] and in previous works [41], our system outperforms ADW (second overall system) by three points (0.80 to 0.77), demonstrating the effectiveness of our direct cross-lingual word comparison with respect to the use of English as a pivot language.

6.1.4. Cross-level semantic similarity

Finally, we evaluated our embedded representations on the word to sense semantic similarity task. Recall from Section 4.2 that our embedded vector representations share the same space with word embeddings. Therefore, in order to calculate the similarity between a word and a sense, we only have to compute the cosine similarity between their respective vector representations.

Dataset. As our benchmark we opted for the *Word to Sense* (word2sense) similarity subtask of the **SemEval-2014 Cross-Level Semantic Similarity** (CLSS) task [57]. The subtask provides 500 word-sense pairs for its test dataset. Each pair is associated with a score denoting the semantic overlap between the two items. From the dataset we took the subset in which the senses are noun instances[29] (277 pairs). This dataset includes many words that are not usually integrated in a knowledge source, such as slang words. Our embedded representations are particularly suitable for this task as they can handle BabelNet Out-Of-Vocabulary words thanks to the shared space of words and senses: if a word is not integrated in BabelNet sense inventory, we simply use the word embedding sharing the same surface form of the given sense.

Comparison systems. Thirty-eight systems participated in the *word2sense* subtask. We compare the performance of our embedded representations with the three best performing participating systems in this subtask. **Meerkat Mafia** [59] is a system that relies on Latent Semantic Analysis (LSA) and uses external dictionaries to handle OOV words. **SemantiKLUE** [113] combines a set of different unsupervised and supervised techniques to measure semantic similarity. The third system, the most similar to our system, is **SimCompass** [81], which relies on deep learning word embeddings and uses WordNet as its only knowledge source.

Results. Table 7 shows Pearson and Spearman correlation performance of the NASARI system with embedded representations together with the three comparison systems. Meerkat Mafia obtains the best overall performance on this dataset. Our system is the second best system, outperforming the remaining 37 participating systems of the SemEval task. Interestingly, NASARI$_{\text{embed}}$ provides a considerable improvement over SimCompass (0.09 and 0.07 in terms of Pearson and Spearman correlations, respectively), which is also based on word embeddings and uses WordNet as lexical resource.

7. Sense Clustering

Our second application focuses on sense clustering. Some sense inventories suffer from a high granularity of their sense inventory. This high granularity could possibly affect the performance of applications based on their sense inventories [103] and, hence, clustering their senses could be beneficial.

Given our setup, we could seamlessly perform sense clustering in BabelNet, WordNet or Wikipedia. We follow the same procedure as semantic similarity for sense clustering. Following [23], we view sense clustering as a binary

[29] Note that our embedded representations can be used to measure the similarity between words with any Part Of Speech tag.

	r	ρ
NASARI$_{\text{embed}}$	0.40	0.40
Meerkat Mafia	**0.44**	**0.44**
SemantiKLUE	0.39	0.39
SimCompass	0.31	0.33

Table 7: Pearson and Spearman correlation performance of different systems on the *word2sense* test set of SemEval-2014 task on Cross-Level Semantic Similarity.

classification task in which given a pair of senses the task is to decide if they have to be merged or not. In the usual setting of clustering, where senses which are semantically related are clustered together, we rely on our similarity scale and simply cluster a pair of items (synsets, senses or pages) together provided that their similarity exceeds the middle point in our similarity scale, i.e., 0.5 in the scale of [0, 1], and with a minimum overlap between vectors of five dimensions. In specific sense clustering settings, this middle-point threshold may be changed to another value, or determined using a tuning dataset.

7.1. Evaluation: Wikipedia sense clustering

Given the high granularity of the Wikipedia sense inventory, clustering related senses may improve systems which take Wikipedia as their knowledge source [46]. Wikipedia-based Word Sense Disambiguation [78, 24] is an example of an application which may benefit from this sense inventory clustering.

7.1.1. Datasets

Wikipedia can be considered as a sense inventory wherein the different meanings of a word are denoted by the articles listed in its disambiguation page [79]. Starting from these Wikipedia disambiguation pages and with the help of human annotation, [23] created two Wikipedia sense clustering datasets. In these datasets, clustering is viewed as a binary classification task in which all possible pairings of senses of a word are annotated whether they should be clustered or not. The first dataset, which we will refer to as **500-pair** dataset, contains 500 pairs, 357 of which are set to belong to the same cluster or *clustered*, and the remaining 143 to *not clustered*. The second dataset, referred to as the **SemEval** dataset, is based on a set of highly ambiguous words taken from SemEval evaluations [78] and consists of 925 pairs, 162 of which are positively labeled (clustered). *Parameter_(computer_programming)-Parameter* and *Fatigue(medical)-Fatigue(safety)* are two sample pairs of Wikipedia pages that should be merged.

As explained above, our system is based on Semantic Similarity (see Section 6) for the sense clustering tasks. Two senses (in this case two Wikipedia pages) are set to be clustered if their similarity is greater than or equal to the middle point of our similarity scale (i.e., 0.5).

7.1.2. Results

Our experiments are carried out on the 500-pair and SemEval datasets. We set two naive baselines: one considering all the pairs as positive or clustered (**Baseline$_{\textbf{cluster}}$**), and another one doing the opposite, i.e., not clustering any of the test pairs (**Baseline$_{\textbf{no-cluster}}$**). We also compare our system to two systems proposed by [23]. Both systems exploit the structure and content of the Wikipedia pages by using a multi-feature Support Vector Machine classifier trained on an automatically-labeled dataset. This first system is totally monolingual (it only makes use of English Wikipedia pages), while the second system also exploits Wikipedia multilinguality[30]. We will refer to the first system as **SVM-monolingual** and to the second system as **SVM-multilingual**.

Table 8 shows the results obtained for the Wikipedia sense clustering task in the 500-pair and SemEval datasets. The results are shown in terms of accuracy (number of correctly labeled pairs divided by total number of instance pairs) and F-Measure (harmonic mean of precision and recall). As we can see from the Table, our system in its unsupervised setting achieves a very high accuracy, outperforming both systems of [23] on the SemEval dataset and

[30]For this second system we report their results for the system configuration which exploits Wikipedia pages in four different languages (English, German, Spanish, and Italian).

Measure	System type	500-pair		SemEval	
		Acc.	F1	Acc.	F1
NASARI	unsupervised	83.8	**70.5**	**87.4**	**63.1**
$\text{NASARI}_{\text{lexical}}$	unsupervised	81.6	65.4	85.7	57.4
$\text{NASARI}_{\text{unified}}$	unsupervised	82.6	69.5	87.2	**63.1**
$\text{NASARI}_{\text{embed}}$	unsupervised	81.2	65.9	86.3	45.5
SVM-monolingual	supervised	77.4	-	83.5	-
SVM-multilingual	supervised	**84.4**	-	85.5	-
$\text{Baseline}_{\text{no-cluster}}$	-	71.4	0.0	82.5	0.0
$\text{Baseline}_{\text{cluster}}$	-	28.6	44.5	17.5	29.8

Table 8: Accuracy (Acc.) and F-Measure (F1) percentages of different systems on the two manually-annotated English Wikipedia sense clustering datasets.

SVM-monolingual on the 500-pair dataset. Only the supervised system of [23] using information of Wikipedia pages in different languages outperforms our main combined NASARI system in terms of accuracy (no F-Measure results were reported) by a narrow margin. Our system, in any of the three variants, comfortably outperforms the naive baselines in terms of both accuracy and F-Measure. When comparing our three systems, the combination of both lexical and unified vectors outperforms both single-handed components. However, both lexical- and unified- based systems (and embedding-based) also prove to be highly competitive single-handed, outperforming all baselines on the SemEval dataset, including the multilingual approach of [23].

8. Domain Labeling

Taking a BabelNet synset (or a Wikipedia page, or a WordNet synset) as input, the task in domain labeling consists of automatically tagging this synset or page with one of the domains in a given set. The domain labeling task has proven to be useful when integrated into a given lexical resource [128, 70] and has several direct applications, such as Word Sense Disambiguation [71, 3, 30] and Text Categorization [95]. WordNet version 3.0 has domains for some of its synsets. However, the number of domains used is quite large (357 domains) and they are not uniformly balanced. For instance, there is a domain named *Ethiopia* containing a single synset, but no other domains referring to different countries are to be found. There are other domains with single synsets, such as *Molecular Biology* or *Cytology*, whereas some domains are annotated with a relatively high number of instances, such as *Law* with 534 annotated instances. Moreover, the coverage of these domains is rather poor: only 4098 synsets have been annotated with at least one domain.

In this section we present a NASARI-based approach to automatically tag a much larger lexical resource (BabelNet) using a different set of domains and achieving significantly higher coverage. Our creation of domain labels for BabelNet synsets relies on our lexical vectors[31]. The first step consists of creating a lexical vector for each domain. To this end, we follow the procedure which was explained in Section 3.2 and learn a lexical vector for each given domain. We do this by using sets of *seed* Wikipedia pages which characterize a given domain. As context for learning the vectors of a given domain we use the concatenations of all the texts corresponding to the Wikipedia pages of the seeds.

Then, in order to find the domain of a synset we computed Weighted Overlap between the corresponding English NASARI lexical vector and the lexical vector of each domain. For a given BabelNet synset s, we pick the domain with maximal similarity:

$$\hat{d}(s) = \underset{d \in D}{\text{argmax}}\ WO(\vec{\text{NASARI}}_{lex}(s), \vec{v}_{lex}(d)) \tag{13}$$

[31] We used lexical vectors because they were shown to perform better for English in Sections 6 and 7.

Animals	Engineering and technology	Language and linguistics	Philosophy and psychology
Art, architecture, and archaeology	Food and drink	Law and Crime	Physics and astronomy
Biology	Games and video games	Literature and theatre	Politics and government
Business, economics, and finance	Geography and places	Mathematics	Religion, mysticism and mythology
Chemistry and mineralogy	Geology and geophysics	Media	Royalty and nobility
Computing	Health and medicine	Meteorology	Sport and recreation
Culture and society	Heraldry, honors, and vexillology	Music	Transport and travel
Education	History	Numismatics and currencies	Warfare and defense

Table 9: Our set of thirty-two domains.

where $\vec{\text{NASARI}}_{lex}(s)$ is the NASARI lexical vector of synset s and $\vec{v}_{lex}(d)$ is the lexical vector of the domain d. Similarly to the sense clustering task, we tagged synsets with a domain provided that the minimum overlap between their respective lexical vectors exceeded five dimensions. For notational brevity, we will refer to the domain of synset s whose score is highest across all domains as its *top domain*.

Wikipedia domains and seeds. To select our set of domains we used Wikipedia featured articles[32], in which a set of 33 domains (e.g. *Animals*, *Meteorology* or *Music*) is provided. For each domain, Wikipedians have selected several Wikipedia pages which best represent that domain. We will refer to these Wikipedia pages already tagged with a domain as *seeds*. Each domain has a different number of available pre-tagged Wikipedia pages, ranging from only 9 (*Mathematics* domain) to 189 (*Media* domain), totalling 4230 Wikipedia pages overall. From the set of domains we decided to remove the *Companies* domain, while retaining the more general *Business, economics, and finance* domain, as we thought *Companies* might conflate with other domains. For instance, *Nestlé* or *Toyota* may be tagged with the *Companies* domain, but also with *Food and drink* and *Transport and travel*, respectively. We also modified some domain names in order to make them more general by taking into account their given seeds. For example, the domain name *Law* was changed to *Law and crime*. The final set of labels includes 32 different domains. Table 9 shows all these domains in alphabetical order.

By applying our pipeline on the Wikipedia seeds over 3.9M BabelNet synsets (from a total of 4.4M English NASARI lexical vectors) were tagged with at least one domain. Over 90% of the 500K synsets that were not annotated with a domain label were isolated Wikipedia pages (i.e., pages that are not linked by any other Wikipedia page) composed of only a few sentences.

8.1. Experiments

In this section we report our experiments on the domain labeling task. First, in Section 8.1.1 we explain the construction of our gold standard domain-labeled datasets. Then, we describe our baseline systems in Section 8.1.2 and compare them against our system on the newly created gold standard datasets in Section 8.1.3.

8.1.1. Gold standard dataset construction

In order to evaluate the performance of our domain labeling approach we constructed two gold standard domain labeled datasets.

WordNet domain-labeled dataset. For the construction of this dataset, we took the WordNet 3.0 synsets which were manually tagged with domains. The domain set of WordNet differs from our set of domains (see Table 9 for our final domain set). Therefore, we performed a manual mapping from the WordNet domains to our domain set in order to make them comparable. Domains in WordNet were mapped to one of our domains provided that the surface form of the WordNet domain matched the surface form of one of our domain labels. For instance, a WordNet synset whose domain was either *Business*, *Economics* or *Finance* was to be mapped to the domain *Business, economics, and finance*. There are WordNet synsets tagged with more than one domain in WordNet, but we considered only those with a single domain in WordNet for the gold standard construction. As a result, we obtained a gold standard dataset of 1540 WordNet synsets tagged with our domain set[33].

[32] `https://en.wikipedia.org/wiki/Wikipedia:Featured_articles`

[33] There is no overlap between these 1540 WordNet synsets and the Wikipedia seeds taken by our system.

BabelNet domain-labeled dataset. In order to have a more realistic distribution of BabelNet synsets comprising not only synsets which belong to the WordNet sense inventory, we created a second gold-standard dataset based on BabelNet. For this, we randomly sampled 200 BabelNet synsets with at least one English lexicalization from the set of all 6.5M possible BabelNet synsets. Of these, 65% were integrated in Wikipedia and only 1.5% belonged to WordNet (the remaining synsets were mostly integrated in WikiData only). Two annotators manually labeled these 200 synsets. They were instructed to mark each synset with a single domain only. Any disagreements were adjudicated in a final phase by the two annotators. The inter-annotator agreement was computed to be 86%, which may be viewed as an upper-bound for the performance of automatic systems.

8.1.2. Comparison systems

As benchmark for our system, we developed three different baselines: two baselines based on Wikipedia and a third one propagating domains using a lexical resource taxonomy. Similarly to our approach, the first two baselines construct a lexical vector for each domain. As seeds, the Wikipedia-based baselines used the same set of Wikipedia pages as our system. Lexical vectors were also computed for each Wikipedia page and the similarity between a Wikipedia page and each domain was calculated. Finally, the Wikipedia page with the top domain similarity score was selected. Vectors were constructed following a classic vector space model scheme in which the resulting vectors are individual content words and the similarity between vectors is calculated by using the standard cosine similarity measure. The only difference between the two baselines lays in the calculation of weights for each dimension. **Wikipedia-TF** calculates weights on the basis of term frequencies (TF), whereas **Wikipedia-TFidf** combines term frequency with the conventional inverse document frequency weighting scheme [52, *tf-idf*]. For these two Wikipedia-based systems, we relied on the mapping provided in BabelNet 3.0 between WordNet synsets and Wikipedia pages.

The third baseline system, **Taxo-Prop** henceforth, uses a taxonomy-based domain propagation. The system takes seeds from the respective domain-labeled gold standard datasets (Section 8.1.1). Algorithm 2 shows the process for obtaining a domain label for a non-tagged synset s. The system is based on a taxonomy and works iteratively. First, it goes over all the neighbours of s in the taxonomy and checks whether they are tagged with a domain (lines 7-16 in the Algorithm). In the case where a particular domain $\hat{d}$ is encountered more often than any other domain among the neighbours' domains, s is tagged with $\hat{d}$ (line 24 in the Algorithm). Otherwise, we repeat the process and move up and down in the taxonomy, thereby checking for the domain tags of the neighbours of the neighbours. We repeat this until a domain appearing more frequently than any other domain is found (lines 5-20 in the Algorithm). In order to test the algorithm on the datasets we used both WordNet **Taxo-Prop (WN)** and BabelNet **Taxo-Prop (BN)** taxonomies and carried out 10-fold cross validation on the test dataset[34].

Finally, we also compared with **WN-Domains-3.2** [70, 9], which is the latest released version of WordNet Domains[35]. The system is in essence very similar to the Taxo-Prop system described above, in the sense that it takes seeds for each domain (manually selected for synsets that are located high in the taxonomy) as input, and spreads them through the WordNet taxonomy. This system involves an undetermined amount of manual intervention in the selection of seeds (*"a small number of high level synsets are manually annotated with their pertinent Subject Code Fields*[36]*"*), and manual curation (*"the main problems are detected and the manual annotations are corrected"*) [70]. WN-Domains-3.2 was released for WordNet 2.0. For testing it on the WordNet-based dataset we used the mapping between versions 2.0 and 3.0 of WordNet[37].

8.1.3. Results

Results are shown in Table 10 in terms of standard precision, recall and F-Measure. When comparing the two Wikipedia-based systems, *tf-idf* proves to be more reliable than using term frequency only, but its performance is still significantly below our NASARI-based system. It is interesting to note that our system is robust across datasets, while Wikipedia-based approaches experience a drastically reduced performance on the BabelNet dataset. This is due to the fact that Wikipedia pages associated with WordNet synsets are, in general, richer and longer than an

[34] In order to make the results more reliable and less sensitive to the dataset order, we repeated this experiment ten times. The gold standard dataset was shuffled each time and the final score was obtained by averaging the results of the ten different runs.

[35] WordNet Domains are available at `http://wndomains.fbk.eu/`

[36] *Subject Code Fields* corresponds to domain labels in our notation.

[37] `https://wordnet.princeton.edu/wordnet/download/current-version/`

Algorithm 2 Taxonomy-based Domain Propagation (Taxo-Prop)

Input: a non-tagged synset s, a set of domain-tagged synsets D, and a function $Tax(s)$ which associates a synset s in the reference sense inventory with the set of its hyponyms and hypernyms in the taxonomy

Output: a domain tag for the input synset s

Set $S \leftarrow \{s\}$
Frequency domain dictionary $F \leftarrow \emptyset$
$tie \leftarrow True$
$S_{prev} \leftarrow \emptyset$
while tie and $|S_{prev}| < |S|$
  $S_{prev} \leftarrow S$
  for each Synset $s' \in S_{prev}$
    for each Neighbour synset $n \in Tax(s')$
      if $n \notin S$ **then**
        $S \leftarrow S \cup \{n\}$
        if $n \in D$ **then**
          Domain $d_n \leftarrow D(n)$
          if $d_n \notin F$ **then**
            $F[d_n] \leftarrow 1$
          else
            $F[d_n] \leftarrow F[d_n] + 1$
  if $|F| > 0$ **then**
    $\hat{d}(s) = \text{argmax}_{d \in F} F[d]$
    if ($|F| = 1$) or ($\max_{d \in F} F[d] > \max_{d \in F \setminus \{\hat{d}(s)\}} F[d]$) **then**
      $tie \leftarrow False$
if tie **then**
  return $null$
else
  return $\hat{d}(s)$

average Wikipedia page (in the BabelNet dataset, synsets were extracted randomly). In contrast, *Taxo-Prop* achieves more competitive results, obtaining a lower precision than our system, but the highest overall recall on the WordNet dataset. However, this may lead to wrong conclusions. Given that the coverage of our system is actually considerably larger than all the synsets covered in WordNet, the recall of our approach is in fact larger than any system relying on WordNet as its only knowledge resource (there are only 117K synsets in the whole of WordNet). Additionally, the WordNet-based approach has the advantage of annotating in exactly the same number of domains as occur in the gold standard dataset. Our system used 4230 Wikipedia pages of 32 different domains as seeds, in contrast to the 1386 domain-labeled WordNet synsets of 27 different domains comprising the gold standard dataset. As a measure to show how much supervision each system was using in each case, we calculated its *seed density*, which is the percentage of synsets used on average for each domain as seeds. Formally, it is calculated as the ratio of the average number of seeds per domain to the total number of synsets in the given resource. In fact, the seed density is significantly higher in the WordNet-based system (0.044% vs. 0.001% of our system).

WN-Domains-3.2 outperforms our system in terms of F-Measure by 2.5 absolute percentage points in the WordNet dataset. Interestingly, despite using as benchmark a subset of WordNet, our system obtains a higher recall than WN-Domains-3.2. Additionally, as remarked above, our system annotates a significantly higher number of instances, including many more named entities and specialized concepts which are not covered by WordNet (over 3.9M domain-labeled synsets annotated by our system as opposed to the 74K synsets annotated by WN-Domains-3.2). In terms of precision, WN-Domains-3.2, which involves an undetermined amount of manual curation, outperforms our default system. However, by simply adding a confidence threshold, our system can considerably increase its precision. For instance, by only tagging synsets whose top domain score is higher than the middle point of our similarity scale (i.e., 0.5), we obtain comparable results in terms of precision percentage (92.5%) to the WN-Domains-3.2 system, while

	WordNet dataset			BabelNet dataset		
	Precision	**Recall**	**F-Measure**	**Precision**	**Recall**	**F-Measure**
NASARI$_{lexical}$	**77.9**	70.1	**73.8**	**62.3**	**40.5**	**49.1**
Wikipedia-TF	25.4	16.4	19.9	3.4	2.5	2.9
Wikipedia-TFidf	45.9	29.7	36.1	8.8	6.5	7.5
Taxo-Prop (WN)	71.3	70.7	71.0	-	-	-
Taxo-Prop (BN)	73.5	**73.5**	73.5	48.3	37.2	42.0
WN-Domains-3.2	93.6	64.4	76.3	-	-	-

Table 10: Precision, Recall and F-Measure percentages of different systems on the gold standard WordNet and BabelNet domain-labeled datasets.

still obtaining a considerably higher coverage.

Note that in both our system and WN-Domains-3.2, the range of domains considered in the original systems was larger than the number of domains found in the gold standard, which increases the error margin. For instance, in the original setting of our system we considered 32 domains (see Table 9), of which only 27 were present in the gold standard dataset. By analyzing the errors given by our system, we realized that there are synsets that might be tagged with more than one domain. If we take the top three domain tags into account, the precision of our system increases to 91.8% and 83.1%, with recall being 82.7% and 54% in the WordNet and the BabelNet datasets, respectively. For example, our system tags the WordNet synset corresponding to the concept *angular_velocity*^1_n with *Mathematics* as top domain by a narrow margin, but in this case it would also be tagged with the *Physics and astronomy domain* as second domain, which would be the right answer according to the gold dataset. As a second source of error, we realized that it is arguable whether many of the false positives given by our system are, in fact, entirely wrong. Indeed, in many cases the judgement made by our system could be considered as justifiable, and equally correct to the tagging found in the gold dataset. For instance, the synset represented by the *data processing* sense of *operation* is tagged with the *Mathematics* domain, while the gold domain is *Computing*. In this case, it is clear that the synset could be tagged with either of the two domains. Another example is the WordNet synset *aesthetics*^1_n, defined in WordNet as *The branch of philosophy dealing with beauty and taste (emphasizing the evaluative criteria that are applied to art)*, which is tagged with the *Philosophy and psychology* domain by our system instead of the *Art, architecture, and archeology* domain label found in the gold dataset.

9. Word Sense Disambiguation

Word Sense Disambiguation (WSD) is a core task in natural language understanding. Given a target word in context, the task consists of associating it with an entry in a given sense repository [94]. WSD may eventually be applied to any Natural Language Processing task, enabling an understanding of the sentences by the machine which is not usually achieved by mainstream statistical approaches, and could benefit applications such as Machine Translation [135] and Information Retrieval [121], to name but a few.

In Section 9.1 we present the different resources which may be used as knowledge repositories for WSD. A unified framework for WSD based on NASARI is presented in Section 9.2. Experiments are presented in Section 9.3.

9.1. Sense inventories

One of the main knowledge sense repositories used in this task was the manually constructed WordNet [96, 112], which usually leads to a fine-grained type of disambiguation given the nature of the senses in WordNet. Another resource more recently used for this task is Wikipedia [79, 24, 96], due to its wide coverage of named entities and multilinguality. A newer resource used as a knowledge repository that is gaining popularity thanks to its multilinguality and large coverage is BabelNet [96, 89, 137], which is our main resource. Given the nature of our vectors, and in contrast to other WSD systems, we can seamlessly disambiguate in any of these resources (BabelNet integrates, among other resources, WordNet and Wikipedia). In the following section, we propose a unified framework for disambiguating words in context irrespective of the resource.

9.2. Framework for Word Sense Disambiguation

In [16] we presented a WSD framework in which we used the lexical vectors and then calculated the overlap between the target word vector and its context, harmonically weighting the ranks of the overlapping words in the target word vector. This method considers each word in context to be equally important (same weight) in the disambiguation process. In this section we present a more suitable approach which keeps to the spirit of previous lexical semantics applications and gives each word its weight in context.

Given a set of target words in a text $\mathcal{T}$, we build a lexical vector for the context, as explained in Section 3.2. Then, for each target word w in the text $\mathcal{T}$, we retrieve the set of all the possible BabelNet synsets which have this target word as one of its lexicalizations, a set we refer to as $\mathcal{L}_w$. Finally, we simply compute Weighted Overlap (see Section 3.5) between $\vec{v}_{lex}(\mathcal{T})$ (the lexical vector of the text $\mathcal{T}$) and the NASARI vectors corresponding to the BabelNet synsets that contain senses of w. In our setting, the top BabelNet synset in terms of WO score ($\hat{s}$) is selected as the best sense of the given target word:

$$\hat{s} = \underset{s \in \mathcal{L}_w}{\operatorname{argmax}} \, WO(\vec{v}_{lex}(\mathcal{T}), \vec{\text{NASARI}}_{lex}(s)) \tag{14}$$

9.3. Experiments

We perform Word Sense Disambiguation experiments using two sense inventories: Wikipedia and WordNet. Recall from Section 9.1 that, since our main knowledge sense inventory is BabelNet, we can seamlessly disambiguate instances using either of these two knowledge sources. The setting of the system is the same in both cases, with only one difference: we use only BabelNet synsets[38] which are mapped to Wikipedia page or WordNet synset when disambiguating with either of these resources, respectively.

As has often been done in the literature [133, 143, 90], we use a back-off strategy to the Most Frequent Sense (MFS) baseline in the cases when our system does not provide a confident answer. Hence, in our WSD framework, we only tagged those instances whose top similarity score (see Section 9.2 for more details on our WSD system) is higher than a given threshold θ. In order to compute θ, we use the English Wikipedia trial dataset provided within the SemEval 2013 WSD task [96]. The top performing value of θ was 0.20, value that is used across all WSD experiments[39].

Section 9.3.1 presents multilingual WSD experiments using Wikipedia as main sense inventory (a task that is strongly related to the *Wikification* task [79]), Section 9.3.2 presents experiments for the Named Entity Disambiguation task using BabelNet as sense inventory, and finally Section 9.3.3 presents the WSD results for English using WordNet as sense inventory.

9.3.1. Multilingual Word Sense Disambiguation using Wikipedia

We used the **SemEval-2013 all-words WSD** dataset [96] as benchmark for our multilingual evaluations[40]. This dataset includes texts for five different languages (English, French, German, Italian and Spanish) with an average of 1303 disambiguated instances per language, including multiword expressions and named entities.

Comparison systems. As comparison system we include **Babelfy** [90][42], a state-of-the-art graph-based system for multilingual joint WSD and Entity Linking. Babelfy relies on random walks in the BabelNet semantic network combined with various graph-based heuristics. We also report results for the best run on every language of the top SemEval-2013 system [40, **UMCC-DLSI**]. As baseline, although difficult to beat in some WSD tasks [94], we include

[38] In order to avoid disambiguating with synsets which are rarely used in practise and are isolated in the BabelNet graph, throughout all the experiments we only considered those BabelNet synsets with at least thirty edges in the BabelNet graph.

[39] We considered values of θ from 0 to 1 with a step size of 0.05.

[40] In our experiments we used the Wikipedia dump of December 2014, as opposed to the one used in the original SemEval 2013 dataset. A few Wikipedia page titles had been updated since the creation of the dataset, so we had to update these titles in the gold standard too[41]. Note that the Wikipedia page titles are the unique identifiers for a Wikipedia page, hence a change in a Wikipedia page title automatically modifies this unique identifier. For instance, the English Wikipedia page titled *Seven-day week* in the SemEval 2013 dataset has been updated in Wikipedia and is currently titled simply *Week*.

[42] `http://babelfy.org/`

System	English	French	Italian	German	Spanish	Average
NASARI$_{lexical}$	86.3	**76.2**	83.7	**83.2**	82.9	**82.5**
MUFFIN	84.5	71.4	81.9	83.1	**85.1**	81.2
Babelfy	**87.4**	71.6	**84.3**	81.6	83.8	81.7
UMCC-DLSI	54.8	60.5	58.3	61.0	58.1	58.5
MFS	80.2	74.9	82.2	83.0	82.1	79.3

Table 11: F-Measure percentage performance on the SemEval-2013 Multilingual WSD datasets using Wikipedia as sense inventory.

the Most Frequent Sense (**MFS**[43]) heuristic. Finally, we report results from **MUFFIN** [16], our previous WSD system based on the NASARI vectors that, in contrast, used a WSD framework in which words in context were considered equally important.

Results. Table 11 shows F-Measure percentage results for our system and all comparison systems on the SemEval 2013 dataset. As we can see from the table, although our system only achieves state-of-the-art results for French and German, it does achieve the best average performance among all languages, demonstrating its robustness across languages and outperforming the current state-of-the-art results of Babelfy. Our system outpeforms our previous WSD approach MUFFIN by over a point on average, highlighting our improvements on this particular WSD task for which we proposed a new framework (see Section 9.2).

9.3.2. English Named Entity Disambiguation using BabelNet

In order to evaluate the quality of our named entity representation, we performed experiments on the Named Entity Disambiguation task. Given that NASARI provides semantic representations for both concepts and named entities, this task was analogous to Word Sense Disambiguation (see Section 9.2) with the only difference being that in this task we only considered entity synsets as candidates. To this end, we used the English named entity dataset from the **All-Words Sense Disambiguation and Entity Linking SemEval 2015** task [89]. This dataset consists of 85 named entities to disambiguate.

Comparison systems. We benchmarked our disambiguation system against the SemEval 2015 top three performing systems, which were the only ones outperforming the MFS baseline: **DFKI** [138], **SUDOKU** [72], and **el92** [119]. DFKI is a multi-objective system based on both global unsupervised and local supervised objectives. SUDOKU uses the Personalized PageRank algorithm after disambiguating monosemous instances within the text. Finally, el92 is based on a weighted voting of various disambiguation systems: Wikipedia Miner [88], TagME [32], DBpedia Spotlight [76], and Babelfy [90].

Results. Table 12 shows F-Measure percentage results on the Named Entity portion of the SemEval 2015 WSD dataset[44]. Our system obtains the second overall position of all seventeen systems that participated in the SemEval 2015 Named Entity Disambiguation task. The combination of global unsupervised and local supervised objectives of DFKI obtains the best overall results. As we show in Section 9.3.3 and discuss in Section 9.4, our system, based solely on global semantic features, generally improves when including local supervised features.

9.3.3. English Word Sense Disambiguation using WordNet

For the task of English WSD using WordNet as main sense inventory, we used two recent SemEval WSD datasets: fine-grained all-words **SemEval-2007** [112] and all-words **SemEval-2013** [96]. We performed experiments on the 162 noun instances of the SemEval-2007 dataset. SemEval-2013's dataset contains 1644 instances.

[43]MFS was provided as baseline by the task organizers. However, the MFS score for French was fixed with respect to [16], which showed a lower MFS F-Measure score. The scorer provided by the organizers was case-sensitive whereas a few Wikipedia page titles in the gold standard file did not match the casing of those in the baseline file, which were all lowercased. This led to misalignments between the gold standard and the baseline file.

[44]We found an inaccuracy in an instance of the gold standard dataset. The unambiguous instance *KAlgebra* is disambiguated with the *KAlgebra* concept in the Catalan language, which belongs to a separate synset of the general *KAlgebra* concept in all languages. This instance is repeated nine times within the dataset. By fixing this issue, our system achieves F-Measure results of over 90%.

System	Type	F-Measure
NASARI$_{lexical}$	unsupervised	87.1
DFKI	supervised	**88.9**
SUDOKU	unsupervised	87.0
el92	systems mix	86.1
MFS	–	85.7

Table 12: F-Measure percentage performance on the English Named Entity Disambiguation dataset from the Multilingual All-Words Sense Disambiguation and Entity Linking SemEval 2015 task using BabelNet as sense inventory.

System	SemEval-2013	SemEval-2007
NASARI$_{lexical}$	66.7	66.7
NASARI$_{lexical}$ +IMS	67.0	**68.5**
MUFFIN	66.0	66.0
Babelfy	65.9	62.7
UKB	61.3	56.0
UMCC-DLSI	64.7	–
Multi-Objective	**72.8**	66.0
IMS	65.3	67.3
MFS	63.2	65.8

Table 13: F-Measure percentage performance on the SemEval-2013 and SemEval-2007 (noun instances) English all-words WSD datatets using WordNet as sense inventory (fine-grained).

Comparison systems. We include the state-of-the-art **IMS** system [144] as a supervised system. As unsupervised systems, we report the performance of two graph-based approaches that are based on random walks over their respective semantic networks: BabelNet [90, **Babelfy**] and WordNet [4, **UKB**]. Another approach that uses BabelNet as reference knowledge base is **Multi-Objective** [137] which views WSD as a multi-objective optimization problem. We also report the results of the best configuration of the top-performing system in the SemEval-2013 dataset, namely **UMCC-DLSI** [40]. As in Section 9.3.1, we also include our earlier WSD system **MUFFIN** for comparison. Finally, we include a system called **NASARI+IMS**, which is based on our WSD framework with the only difference being that in this system we back-off to IMS instead of MFS[45].

Results. Table 13 shows the F-Measure percentage performance of all systems on the SemEval-2007 and SemEval-2013 WSD datasets. Similarly to the WSD results using Wikipedia as main sense inventory (Section 9.3.1), our system NASARI outperforms our previous MUFFIN system. NASARI in its default setting backing-off to MFS is only surpassed by Multi-Objective in SemEval-2013 and IMS in SemEval-2017, outperforming the remaining systems in both datasets.

Our system backing-off to IMS (NASARI+IMS) improves our default NASARI system in both datasets, obtaining the best performance among all systems on the SemEval-2007 dataset. We remark that NASARI is an unsupervised system based on global contexts, while IMS is a supervised system based on local contexts. This combination of local and global contexts has already shown to be beneficial for WSD tasks [45, 108, 137].

9.4. Discussion: global and local contexts

Our method is based on global contexts (we use the whole text as context of the target word to disambiguate), hence it sometimes fails to capture the correct meaning of the word in the cases where the local context appears to be the key to the disambiguation, especially in a fine-grained disambiguation scheme. For instance, in the following sentence taken from the SemEval 2013 Word Sense Disambiguation test set, we find an example where a fine-grained distinction of the target word *behaviour* leads to a mistake by our method which could be solved by exploiting the local context by a supervised system:

[45]The MFS baseline was obtained from the SemCor sense-annotated corpus [86].

(1) The expulsion presumably forged by two players of Real Madrid (Xabi Alonso and Sergio Ramos) in the game played on the 23rd of November against Ajax in European Champions League has caused rivers of ink to be written about if such *behaviour* is or is not unsportmanlike and if, both players should be sanctioned by UEFA.

Our system is not confident enough and hesitates between the sense behaviour^3_n (*The aggregate of the responses or reactions or movements made by an organism in any situation*) and behaviour^4_n (*Manner of acting or controlling yourself*), selecting the latter by a narrow margin. In this case, combining our method with one exploiting local contexts such as IMS would lead to the correct answer.

On the other hand, there are cases where a local-based approach may fail due to the lack of a more global text understanding. We appreciate this phenomenon in the following sentence, also taken from the SemEval 2013 dataset:

(2) This way, and since Real Madrid will finish as leader of its group, both players will fulfil the prescribed *sanction* during the next game of league.

In this case, IMS considers as its highest confidence sense sanction^1_n (*Formal and explicit approval*), which is also the most frequent sense for the noun *sanction*. It gets misled by the closest context and would need to get the higher picture (global context) to fix the error. In this case, NASARI correctly captures the semantics within the text and chooses sanction^2_n (*A mechanism of social control for enforcing a society's standards*).

In both cases the combination of NASARI and IMS gets to the correct answer and in general the combination of both methods shows a consistent improvement over the single system components. In fact, the results of the combination of a knowledge-based global-context disambiguation system (i.e., NASARI) with a state-of-the-art supervised local-context approach (i.e., IMS) proves to be quite robust across datasets, outperforming many strong baselines as we can see from Table 13.

10. Analysis

In order to gain a better insight into the role some of the key components of our system's pipeline play in the overall performance, we carried out an ablation test. In particular, we were interested in evaluating the impact and importance of the following three components:

1. **Lexical specificity.** To check how lexical specificity (see Section 3.1) fares against the standard *tf-idf* measure [52], we generated NASARI lexical vectors in which weights were calculated using the conventional *tf-idf*. Given a word w, we calculate $TFidf(w)$ as follows:

$$TFidf(w) = f(w) \log \frac{|D|}{|\{p \in D : w \in p\}|} \tag{15}$$

   where $f(w)$ is the frequency of w in the subcorpus $\mathcal{SC}_s$ representing the contextual information of the synset s (see Section 4.1) and D is the set of all pages in Wikipedia. We computed two sets of *tf-idf*-based lexical vectors. The first version, called NASARI-TFidf, keeps all the dimensions in the vector. For the second version, NASARI-TFidf-3000d, we follow [37] and prune the vector to its top 3000 non-zero dimensions. This pruning is similar to the one performed automatically by lexical specificity, which reduces the number of non-zero dimensions while retaining the interpretability of the vector dimensions.
2. **Weighted semantic relations.** To assess the advantage we gain from introducing weights to semantic relations (see Section 4.1.1), we computed a version of our lexical vectors in which the semantic relations were uniformly weighted (i.e., $\lambda_i = 1, \forall i \in \{1, \ldots, n\}$ in Equation 11), as was the case in our earlier work [16]. We will refer to this version as NASARI-unif.weight.
3. **Combination strategy of embeddings.** Finally, we carried out an analysis to compare the harmonic combination of word embeddings (see Section 3.3) against uniform combination (i.e., averaging). For this purpose, we computed the embedding vector for a given synset as the centroid of all the embeddings of the words present in its corresponding lexical vector. We will refer to this variant as NASARI-av.embed in our tables.

We evaluated the first two components in an intrinsic task (word similarity) as well as a downstream application (Word Sense Disambiguation). For the third component, we compared our default NASARI$_{\text{embed}}$ and the embedding

	Word Similarity						Word Sense Disambiguation	
	MC-30		WS-Sim		SL-999 (nouns)		SemEval-2007	SemEval-2013
	r	ρ	r	ρ	r	ρ	F-Measure	F-Measure
NASARI$_{\text{lexical}}$	**0.88**	**0.81**	**0.74**	**0.73**	**0.51**	**0.49**	**66.7**	**66.7**
NASARI-TFidf	0.84	0.77	0.71	0.71	0.46	0.46	66.0	66.1
NASARI-TFidf-3000d	0.85	0.79	0.72	0.72	0.48	0.47	66.0	65.9
NASARI-unif.weight	0.86	0.79	0.73	0.72	0.49	0.48	66.0	66.4
NASARI$_{\text{embed}}$	**0.91**	**0.83**	**0.68**	**0.68**	**0.48**	**0.46**	**–**	**–**
NASARI-av.embed	0.81	0.75	0.58	0.63	0.40	0.41	–	–

Table 14: Ablation test. Pearson (r) and Spearman (ρ) correlations on RG-65, MC-30, WS-Sim and SimLex-999 (noun instances) word similarity datasets (columns 2-7). F-Measure percentage performance on the SemEval-2007 and SemEval-2013 Word Sense Similarity datasets using WordNet as sense inventory (columns 8-9).

representations obtained through uniform weighting in the word similarity task. We performed the evaluations on the same datasets as those used in Section 6.1.1 for word similarity and in Section 9.3.3 for Word Sense Disambiguation with WordNet as sense inventory. The whole pipeline for both tasks was left unchanged for all variants, except for the components mentioned above.

Table 14 shows the results of the ablation test on Word Similarity and Word Sense Disambiguation. Our default NASARI$_{\text{lexical}}$ system consistently outperforms all baselines in all datasets of both tasks, demonstrating the reliability of the proposed lexical specificity and the preweighting of the semantic relations. This result is especially meaningful taking into account that our default system is the one with the fewest non-zero dimensions on average among the four evaluated approaches. In fact, the average number of non-zero dimensions of our NASARI$_{\text{lexical}}$ vectors was 162, which is lower than the 280 non-zero dimensions of NASARI-unif.weight, 1033 of NASARI-TFidf-3000d[46], and 1561 of NASARI-TFidf. This low average number of non-zero dimensions enables a fast processing of the vectors, i.e., they are computationally faster to work with.

As far as the NASARI$_{\text{embed}}$ vectors are concerned, our default system consistently obtained significantly better results when compared to the baseline (NASARI-av.embed). In general, NASARI-av.embed produces consistently high similarity values, even for non-similar pairs. This is due to the fact that words that are not very relevant to the input synset (i.e., relatively low lexical specificity values) are given the same weight as words that are clearly more relevant (i.e., high lexical specificity values). This, in turn, is why a weighted average of the word embeddings in the lexical vector leads to more accurate results than a simple average.

11. Related Work

In addition to the semantic representation of word senses, which is the main topic of this article, we briefly review the recent literature on the two most popular applications on which we evaluated our representations: semantic similarity and Word Sense Disambiguation.

11.1. Representation of word senses

Most research studies in semantic representation have so far concentrated on the representation of words, as can be seen from the numerous available word similarity datasets and benchmarks, while relatively few studies have focused on the representation of word senses or concepts. This is partly due to the so-called knowledge acquisition bottleneck that arises because the application of distributional word modeling techniques (which are the prominent representation approach) at the sense level would require the availability of high-coverage sense-annotated data. However, word

[46] In NASARI-TFidf-3000d the maximum number of non-zero dimensions is set to 3000, but in many cases the vector has actually a lower number of non-zero dimensions.

representations are known to suffer from some issues which dampen their suitability for tasks that require accurate representations of meaning. The most important drawback with word representations lies in their inability to model polysemy and homonymy, as they conflate different meanings that a word can have into a single representation [131, 115]. For instance, a word representation for the word *bank* does not distinguish between the financial institution and the river bank meanings of the word (the noun *bank* has ten senses according to WordNet 3.0). The approach of [31] which leverages semantic lexicons to improve word representations also suffers from the same drawback.

Because they represent the lowest linguistic level, word senses and concepts play a crucial role in natural language understanding. Since at this level individual meanings of a word are identified and separately modeled, the resulting representations are ideal for accurate semantic representation. In addition, the fine-grained representation of word senses can be directly extended to higher linguistic levels [13], such as words, which makes them quite interesting. These features have recently attracted the attention of different research studies. Most of these techniques view sense representation as a specific type of word representation and try to adapt the existing distributional word modeling techniques to the sense level, usually through clustering the contexts in which a word appears [139, 47, 100]. The fundamental assumption here is that the intended meaning of a word mainly depends on its context and hence one can obtain sense-specific contexts for a given word sense by clustering the contexts in which the word appears in a given text corpus. Various clustering-based techniques usually differ in their clustering procedure and how this is combined with the representation technique. However, these models are often limited to representing only those senses that are covered in the underlying corpus. Moreover, the sense representations obtained using these methods are usually not linked to any sense inventory, and therefore such linking has to be carried out, either manually, or with the help of sense-annotated data if the representations are to be used for direct applications such as Word Sense Disambiguation.

Most sense modeling techniques have based their representation on the knowledge derived from resources such as WordNet. Earlier techniques exploit the information provided in WordNet, such as the synonymous words in a synset, for the representation of word senses [80, 2]. More recent approaches usually adapt distributional models to the sense level on the basis of lexico-semantic knowledge derived from lexical resources such as Wikipedia [35, 78], WordNet [19, 50, 117] or other language-specific semantic networks [51]. WordNet is also viewed as a semantic network where its individual synsets are represented on the basis of graph-based algorithms [107]. Word Sense Disambiguation of large amounts of textual data has also been explored as a means of obtaining high-coverage annotated data for learning sense representations based on neural networks, a representation referred to as sense embeddings [48]. [19], which uses WordNet as main knowledge source, also relies on WSD for obtaining their sense representations. However, these two approaches are hampered by their inherently imperfect WSD systems.

Additionally, these techniques are often limited to the reduced coverage of WordNet and to the English language only. In contrast, our method provides a multilingual representation of word senses on the basis of the complementary knowledge of two different resources, enabling a significantly higher coverage of specific domains and named entities. Our representations are not only multilingual, but can also be compared across languages through our unified representations.

11.2. Semantic similarity

Semantic similarity between word senses is usually computed on the basis of the structural properties of lexical databases such as WordNet [6, 13], or thesauri such as Roget's [91, 49]. These measures often represent a lexical resource as a semantic network and then exploit the networks for the computation of semantic similarity between a pair of word senses. The conventional WordNet-based similarity techniques take as their source of information either only the structural properties of the WordNet semantic network, such as graph distance and the lowest common super-ordinate of two word senses [44, 65, 140], or combine the structural information with statistics obtained from text corpora [116, 69]. Collaboratively-constructed resources such as Wikipedia and Wiktionary have also been used as underlying lexical resources in different semantic similarity techniques [41, 127, 87]. More recent sense similarity methods first perform random walks on the semantic networks [107, 142, 110] in order to model individual word senses and then use these representations for the computation of sense similarity. All these techniques, however, are limited to the knowledge provided by their underlying semantic resource. In contrast, our approach combines expert-based and encyclopedic knowledge from two different types of resource, providing three advantages: (1) more effective measurement of similarity based on rich semantic representations, (2) the possibility of measuring cross-resource semantic similarity, i.e., between Wikipedia pages and WordNet synsets, and (3) the possibility of comparing the semantics of word senses across different languages.

11.3. Word Sense Disambiguation

Word Sense Disambiguation is a task that can benefit significantly from the representation of word senses, mainly due to its sense-level application. Based on the type of resources they use, WSD techniques can be put into two main categories: knowledge-based and supervised [94]. Supervised systems receive sense-annotated data as their source of information, i.e., a set of contexts in which a specific sense of a word appears. These systems analyze the provided data and capture the context in which a specific word sense is more likely to appear. It Makes Sense [144, IMS] is an example of a supervised system which, despite using a small set of conventional features and a simple linear classifier, has been among the best performers on different WSD bechmarks. However, the performance of supervised systems very much depends on the availability of sense-annotated data for the target word sense [108]. Hence, the applicability of these systems is limited to those words and languages for which such data is available, practically restricting them to a small subset of word senses and mainly for the English language only. Knowledge-based approaches, on the other hand, do not suffer from the lack of sense-annotated data and therefore provide a relatively higher coverage. These systems usually exploit the structural or lexical-semantic information in lexical resources for disambiguation [123, 97, 4]. However, similarly to their supervised counterparts, knowledge-based techniques are mostly limited to the English language only. Recent years have seen a growing interest in multilingual WSD [96]. Multilinguality is usually offered by methods that exploit the structural information of large-scale multilingual lexical resources such as Wikipedia [40, 73, 46]. Babelfy [90] is such a WSD system which performs random walks on the BabelNet multilingual semantic network [99] and makes use of densest subgraph heuristics. However, the approach is limited to the WSD and Entity Linking tasks. In contrast, our approach is global, as it can be used in different NLP tasks, including WSD and Entity Linking.

12. Conclusions

In this article we presented NASARI, a novel technique for the representation of concepts and named entities in arbitrary languages. Our approach combines the structural knowledge from semantic networks with the statistical information derived from text corpora for effective representation of millions of BabelNet synsets, including WordNet nominal synsets and all Wikipedia pages. We evaluated our representations in a wide range of NLP tasks and applications: semantic similarity, sense clustering, Word Sense Disambiguation, and domain labeling. We reported state-of-the-art performance on several datasets across these tasks and in different languages.

Three type of sense representation were put forward: two explicit vector representations (unified and lexical) in which vector dimensions are interpretable and a latent embedding-based representation. Each representation has its own advantages and disadvantages. In general, a combination of lexical and unified vectors led to the most reliable results in the semantic similarity and sense clustering experiments (Sections 6 and 7). Among the three representations, the lexical representation (i.e., NASARI$_{\text{lexical}}$) obtained the best performance in monolingual settings. However, although the lexical vectors are sparse and computationally easy to work with in many applications, the dimensionality is high as it is equal to the vocabulary size. In contrast, our embedded representation (i.e., NASARI$_{\text{embed}}$) has a fixed low number of latent dimensions. Additionally, embedded synset vectors share the same space with the word embeddings used as input. As regards our unified representation (i.e., NASARI$_{\text{unified}}$), not only does it provide an effective way for representing word senses in different languages, but, thanks to its unified semantic space, it also enables a direct comparison of different representations across languages. In addition to being multilingual, NASARI improves over the existing techniques by providing a high coverage for millions of concepts and named entities defined in the BabelNet sense inventory.

Release. We are releasing the complete set of representations obtained using our technique for five different languages (English, Spanish, French, German and Italian) at `http://lcl.uniroma1.it/nasari`, and we plan to generate representations for more languages in the near future. We also provide a Python script for fast computation of lexical specificity. Additionally, domain labels are also included in the BabelNet 3.5 release version[47]. We also release the gold standard domain-labeled datasets used for our experiments (Section 8.1.1).

[47] BabelNet domain labels are based on NASARI and have been extended by using a set of taxonomy-based heuristics. BabelNet 3.5 includes 1.65M synsets annotated with at least one domain label.

Future work. As future work we plan to pursue three main directions. Firstly, we aim to compute a global representation for each concept by exploiting the statistical information obtained from multiple languages. Secondly, we plan to develop a framework for a more meaningful combination of our representations in a supervised system for improved joint WSD and Entity Linking. Thirdly, we plan to integrate our multilingual semantic representations into different end-user applications, such as Machine Translation.

Acknowledgments

The authors gratefully acknowledge the support of the ERC Starting Grant MultiJEDI No. 259234.

13. Bibliography

[1] Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Paşca, M., Soroa, A., 2009. A study on similarity and relatedness using distributional and WordNet-based approaches. In: Proceedings of NAACL. pp. 19–27.

[2] Agirre, E., de Lacalle, O. L., 2004. Publicly available topic signatures for all WordNet nominal senses. In: Proceedings of LREC. Lisbon, Portugal, pp. 1123–1126.

[3] Agirre, E., de Lacalle, O. L., Soroa, A., 2009. Knowledge-based WSD on specific domains: performing better than generic supervised WSD. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI). Pasadena, California, pp. 1501–1506.

[4] Agirre, E., Soroa, A., 2009. Personalizing PageRank for Word Sense Disambiguation. In: Proceedings of EACL. pp. 33–41.

[5] Al-Rfou, R., Perozzi, B., Skiena, S., 2013. Polyglot: Distributed word representations for multilingual nlp. In: Proceedings of the Seventeenth Conference on Computational Natural Language Learning. Sofia, Bulgaria, pp. 183–192.

[6] Banerjee, S., Pedersen, T., 2002. An adapted Lesk algorithm for Word Sense Disambiguation using WordNet. In: Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing. CICLing'02. Mexico City, Mexico, pp. 136–145.

[7] Bär, D., Zesch, T., Gurevych, I., August 2013. DKPro similarity: An open source framework for text similarity. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations. Sofia, Bulgaria, pp. 121–126.

[8] Baroni, M., Dinu, G., Kruszewski, G., 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In: Proceedings of ACL. pp. 238–247.

[9] Bentivogli, L., Forner, P., Magnini, B., Pianta, E., 2004. Revising the wordnet domains hierarchy: semantics, coverage and balancing. In: Proceedings of the Workshop on Multilingual Linguistic Ressources. Association for Computational Linguistics, pp. 101–108.

[10] Billami, M.-B., Camacho-Collados, J., Jacquey, E., Kister, L., 2014. Annotation sémantique et validation terminologique en texte intégral en SHS. In: Proceedings of TALN. pp. 363–376.

[11] Bordag, S., 2006. Word sense induction: Triplet-based clustering and automatic evaluation. In: Proceedings of the 11^{th} Conference on European chapter of the Association for Computational Linguistics (EACL). Trento, Italy, pp. 137–144.

[12] Brody, S., Lapata, M., 2009. Bayesian Word Sense Induction. In: Proceedings of EACL. pp. 103–111.

[13] Budanitsky, A., Hirst, G., 2006. Evaluating WordNet-based measures of Lexical Semantic Relatedness. Computational Linguistics 32 (1), 13–47.

[14] Camacho-Collados, J., Billami, M., Jacquey, E., Kister, L., 2014. Approche statistique pour le filtrage terminologique des occurrences de candidats termes en texte intégral. In: Proceedings of JADT. pp. 121–133.

[15] Camacho-Collados, J., Pilehvar, M. T., Navigli, R., 2015. A Framework for the Construction of Monolingual and Cross-lingual Word Similarity Datasets. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing – Short Papers. Beijing, China, pp. 1–7.

[16] Camacho-Collados, J., Pilehvar, M. T., Navigli, R., 2015. 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. Beijing, China, pp. 741–751.

[17] Camacho-Collados, J., Pilehvar, M. T., Navigli, R., 2015. NASARI: a Novel Approach to a Semantically-Aware Representation of Items. In: Proceedings of NAACL. pp. 567–577.

[18] Cardellino, C., March 2016. Spanish Billion Words Corpus and Embeddings. URL `http://crscardellino.me/SBWCE/`

[19] Chen, X., Liu, Z., Sun, M., 2014. A unified model for word sense representation and disambiguation. In: Proceedings of EMNLP. Doha, Qatar, pp. 1025–1035.

[20] Collobert, R., Weston, J., 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of ICML. pp. 160–167.

[21] Crouch, C. J., 1988. A cluster-based approach to thesaurus construction. In: Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '88. pp. 309–320.

[22] Curran, J. R., Moens, M., 2002. Improvements in automatic thesaurus extraction. In: Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition - Volume 9. ULA '02. pp. 59–66.

[23] Dandala, B., Hokamp, C., Mihalcea, R., Bunescu, R. C., 2013. Sense clustering using Wikipedia. In: Proceedings of Recent Advances in Natural Language Processing. Hissar, Bulgaria, pp. 164–171.

[24] Dandala, B., Mihalcea, R., Bunescu, R., 2013. Word sense disambiguation using wikipedia. In: The Peoples Web Meets NLP. Springer, pp. 241–262.
[25] Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., Harshman, R. A., 1990. Indexing by latent semantic analysis. Journal of American Society for Information Science 41 (6), 391–407.
[26] Delli Bovi, C., Telesca, L., Navigli, R., 2015. Large-scale information extraction from textual definitions through deep syntactic and semantic analysis. Transactions of the Association for Computational Linguistics 3, 529–543.
[27] Di Marco, A., Navigli, R., 2013. Clustering and diversifying web search results with graph-based word sense induction. Computational Linguistics 39 (3), 709–754.
[28] Drouin, P., 2003. Term extraction using non-technical corpora as a point of leverage. Terminology 9 (1), 99–115.
[29] Erk, K., 2007. A simple, similarity-based model for selectional preferences. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Prague, Czech Republic. pp. 216–223.
[30] Faralli, S., Navigli, R., 2012. A New Minimally-supervised Framework for Domain Word Sense Disambiguation. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Jeju, Korea, pp. 1411–1422.
[31] Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., Smith, N. A., 2015. Retrofitting word vectors to semantic lexicons. In: Proceedings of NAACL. pp. 1606–1615.
[32] Ferragina, P., Scaiella, U., 2010. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In: Proceedings of the 19th ACM international conference on Information and knowledge management. ACM, pp. 1625–1628.
[33] Finkelstein, L., Evgenly, G., Yossi, M., Ehud, R., Zach, S., Gadi, W., Eytan, R., 2002. Placing search in context: The concept revisited. ACM Transactions on Information Systems 20 (1), 116–131.
[34] Flati, T., Vannella, D., Pasini, T., Navigli, R., 2014. Two is bigger (and better) than one: the Wikipedia Bitaxonomy project. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Baltimore, USA, pp. 945–955.
[35] Gabrilovich, E., Markovitch, S., 2007. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In: Proceedings of IJCAI. pp. 1606–1611.
[36] Gärdenfors, P., 2004. Conceptual spaces: The geometry of thought. The MIT Press.
[37] Gong, Y., Wang, L., Hodosh, M., Hockenmaier, J., Lazebnik, S., 2014. Improving image-sentence embeddings using large weakly annotated photo collections. In: Computer Vision–ECCV 2014. Springer, pp. 529–545.
[38] Granada, R., Trojahn, C., Vieira, R., 2014. Comparing semantic relatedness between word pairs in Portuguese using Wikipedia. In: Computational Processing of the Portuguese Language. pp. 170–175.
[39] Gurevych, I., 2005. Using the structure of a conceptual network in computing semantic relatedness. In: Proceedings of IJCNLP. pp. 767–778.
[40] Gutiérrez, Y., Castañeda, Y., González, A., Estrada, R., Piug, D. D., Abreu, I. J., Pérez, R., Fernández Orquín, A., Montoyo, A., Muñoz, R., Camara, F., 2013. UMCC_DLSI: Reinforcing a ranking algorithm with sense frequencies and multidimensional semantic resources to solve multilingual word sense disambiguation. In: Proceedings of SemEval 2013. pp. 241–249.
[41] Hassan, S., Mihalcea, R., 2011. Semantic relatedness using salient semantic analysis. In: Proceedings of AAAI. pp. 884–889.
[42] Heiden, S., Magué, J.-P., Pincemin, B., et al., 2010. Txm: Une plateforme logicielle open-source pour la textométrie-conception et développement. In: Statistical Analysis of Textual Data-Proceedings of 10th International Conference Journées d'Analyse statistique des Données Textuelles. Vol. 2. Rome, Italy, pp. 1021–1032.
[43] Hill, F., Reichart, R., Korhonen, A., 2014. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. ArXiv:1408.3456.
[44] Hirst, G., St-Onge, D., 1998. Lexical chains as representations of context for the detection and correction of malapropisms. In: Fellbaum, C. (Ed.), WordNet: An electronic lexical database. MIT Press, pp. 305–332.
[45] Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., Weikum, G., 2011. Robust disambiguation of named entities in text. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, pp. 782–792.
[46] Hovy, E. H., Navigli, R., Ponzetto, S. P., 2013. Collaboratively built semi-structured content and Artificial Intelligence: The story so far. Artificial Intelligence 194, 2–27.
[47] Huang, E. H., Socher, R., Manning, C. D., Ng, A. Y., 2012. Improving word representations via global context and multiple word prototypes. In: Proceedings of ACL. Jeju Island, South Korea, pp. 873–882.
[48] Iacobacci, I., Pilehvar, M. T., Navigli, R., 2015. Sensembed: Learning sense embeddings for word and relational similarity. 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). Association for Computational Linguistics, Beijing, China, pp. 95–105.
[49] Jarmasz, M., Szpakowicz, S., 2003. Roget's thesaurus and semantic similarity. In: Proceedings of RANLP. pp. 212–219.
[50] Jauhar, S. K., Dyer, C., Hovy, E., 2015. Ontologically grounded multi-sense representation learning for semantic vector space models. In: Proceedings of NAACL. pp. 683–693.
[51] Johansson, R., Pina, L. N., 2015. Embedding a semantic network in a word space. In: Proceedings of NAACL. pp. 1428–1433.
[52] Jones, K. S., 1972. A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation 28, 11–21.
[53] Jones, M. P., Martin, J. H., 1997. Contextual spelling correction using latent semantic analysis. In: Proceedings of the Fifth Conference on Applied Natural Language Processing. ANLC '97. pp. 166–173.
[54] Joubarne, C., Inkpen, D., 2011. Comparison of semantic similarity for different languages using the Google n-gram corpus and second-order co-occurrence measures. In: Advances in Artificial Intelligence. pp. 216–221.
[55] Jurgens, D., 2013. Embracing ambiguity: A comparison of annotation methodologies for crowdsourcing word sense labels. In: HLT-NAACL. pp. 556–562.
[56] Jurgens, D., Navigli, R., 2014. It's All Fun and Games until Someone Annotates: Video Games with a Purpose for Linguistic Annotation. Transactions of the Association for Computational Linguistics (TACL) 2, 449–464.
[57] Jurgens, D., Pilehvar, M. T., Navigli, R., 2014. Semeval-2014 task 3: Cross-level semantic similarity. SemEval 2014, 17–26.
[58] Jurgens, D., Stevens, K., 2011. Measuring the impact of sense similarity on Word Sense Induction. In: Proceedings of the First Workshop

on Unsupervised Learning in NLP. EMNLP '11. Edinburgh, Scotland, pp. 113–123.
[59] Kashyap, A., Han, L., Yus, R., Sleeman, J., Satyapanich, T., Gandhi, S., Finin, T., 2014. Meerkat mafia: Multilingual and cross-level semantic textual similarity systems. In: Proceedings of the 8th International Workshop on Semantic Evaluation. Association for Computational Linguistics, pp. 416–423.
[60] Kennedy, A., Hirst, G., 2012. Measuring semantic relatedness across languages. In: Proceedings of xLiTe: Cross-Lingual Technologies Workshop at the Neural Information Processing Systems Conference.
[61] Laender, A. H. F., Ribeiro-Neto, B. A., da Silva, A. S., Teixeira, J. S., 2002. A brief survey of web data extraction tools. SIGMOD Rec. 31 (2), 84–93.
[62] Lafon, P., 1980. Sur la variabilité de la fréquence des formes dans un corpus. Mots 1, 127–165.
[63] Landauer, T., Dooley, S., 2002. Latent semantic analysis: theory, method and application. In: Proceedings of CSCL. pp. 742–743.
[64] Landauer, T. K., Dumais, S. T., 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological Review 104 (2), 211–240.
[65] Leacock, C., Chodorow, M., 1998. Combining local context and WordNet similarity for word sense identification. In: Fellbaum, C. (Ed.), WordNet: An electronic lexical database. MIT Press, pp. 265–283.
[66] Lebart, L., Salem, A., Berry, L., 1998. Exploring textual data. Kluwer Academic Publishers.
[67] Levy, O., Goldberg, Y., Dagan, I., 2015. Improving distributional similarity with lessons learned from word embeddings. Transactions of the Association for Computational Linguistics 3, 211–225.
[68] Li, J., Jurafsky, D., 2015. Do multi-sense embeddings improve natural language understanding? In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP). Lisbon, Portugal, pp. 1722–1732.
[69] Lin, D., 1998. An information-theoretic definition of similarity. In: Proceedings of the Fifteenth International Conference on Machine Learning. San Francisco, CA, pp. 296–304.
[70] Magnini, B., Cavaglia, G., 2000. Integrating subject field codes into WordNet. In: LREC. pp. 1413–1418.
[71] Magnini, B., Strapparava, C., Pezzulo, G., Gliozzo, A., 2002. The role of domain information in word sense disambiguation. Natural Language Engineering 8 (04), 359–373.
[72] Manion, S. L., 2015. Sudoku: Treating word sense disambiguation & entity linking as a deterministic problem–via an unsupervised & iterative approach. 9th International Workshop on Semantic Evaluation (SemEval 2015), 365.
[73] Manion, S. L., Sainudiin, R., 2013. Daebak!: Peripheral diversity for multilingual Word Sense Disambiguation. In: Proceedings of SemEval 2013. pp. 250–254.
[74] Matuschek, M., Gurevych, I., 2013. Dijkstra-WSA: A graph-based approach to word sense alignment. Transactions of the Association for Computational Linguistics (TACL) 1, 151–164.
[75] McCarthy, D., Navigli, R., 2009. The English lexical substitution task. Language Resources and Evaluation 43 (2), 139–159.
[76] Mendes, P. N., Jakob, M., García-Silva, A., Bizer, C., 2011. Dbpedia spotlight: shedding light on the web of documents. In: Proceedings of the 7th International Conference on Semantic Systems. ACM, pp. 1–8.
[77] Mihalcea, R., 2007. Using Wikipedia for automatic Word Sense Disambiguation. In: Proceedings of NAACL-HLT-07. Rochester, NY, pp. 196–203.
[78] Mihalcea, R., 2007. Using Wikipedia for automatic Word Sense Disambiguation. In: Proc. of NAACL-HLT-07. Rochester, NY, pp. 196–203.
[79] Mihalcea, R., Csomai, A., 2007. Wikify! Linking documents to encyclopedic knowledge. In: Proceedings of the Sixteenth ACM Conference on Information and Knowledge management. Lisbon, Portugal, pp. 233–242.
[80] Mihalcea, R., Moldovan, D., 1999. An automatic method for generating sense tagged corpora. In: Proceedings AAAI '99. Orlando, Florida, USA, pp. 461–466.
[81] Mihalcea, R., Wiebe, J., 2014. Simcompass: Using deep learning word embeddings to assess cross-level similarity. SemEval 2014, 560.
[82] Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. CoRR abs/1301.3781. URL `http://arxiv.org/abs/1301.3781`
[83] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J., 2013. Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119.
[84] Miller, G. A., Beckwith, R., Fellbaum, C. D., Gross, D., Miller, K., 1990. WordNet: an online lexical database. International Journal of Lexicography 3 (4), 235–244.
[85] Miller, G. A., Charles, W. G., 1991. Contextual correlates of semantic similarity. Language and Cognitive Processes 6 (1), 1–28.
[86] Miller, G. A., Leacock, C., Tengi, R., Bunker, R., 1993. A semantic concordance. In: Proceedings of the 3rd DARPA Workshop on Human Language Technology. Plainsboro, N.J., pp. 303–308.
[87] Milne, D., Witten, I. H., 2008. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In: Proceedings of the Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy at AAAI-08. Chicago, IL, pp. 25–30.
[88] Milne, D., Witten, I. H., 2008. Learning to link with Wikipedia. In: Proc. of CIKM-08. pp. 509–518.
[89] Moro, A., Navigli, R., 2015. Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. Proceedings of SemEval-2015, 288–297.
[90] Moro, A., Raganato, A., Navigli, R., 2014. Entity Linking meets Word Sense Disambiguation: a Unified Approach. Transactions of the Association for Computational Linguistics (TACL) 2, 231–244.
[91] Morris, J., Hirst, G., 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. Computational Linguistics 17 (1), 21–43.
[92] Mrkšić, N., Séaghdha, D. Ó., Thomson, B., Gašić, M., Rojas-Barahona, L., Su, P.-H., Vandyke, D., Wen, T.-H., Young, S., 2016. Counter-fitting word vectors to linguistic constraints. arXiv preprint arXiv:1603.00892.
[93] Navigli, R., 2006. Meaningful clustering of senses helps boost word sense disambiguation performance. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics. pp. 105–112.
[94] Navigli, R., 2009. Word Sense Disambiguation: A survey. ACM Computing Surveys 41 (2), 1–69.

[95] Navigli, R., Faralli, S., Soroa, A., de Lacalle, O., Agirre, E., 2011. Two birds with one stone: Learning semantic models for text categorization and Word Sense Disambiguation. In: Proceedings of the 20th ACM Conference on Information and Knowledge Management (CIKM). Glasgow, UK, pp. 2317–2320.

[96] Navigli, R., Jurgens, D., Vannella, D., 2013. SemEval-2013 Task 12: Multilingual Word Sense Disambiguation. In: Proceedings of SemEval 2013. pp. 222–231.

[97] Navigli, R., Lapata, M., 2007. Graph connectivity measures for unsupervised Word Sense Disambiguation. In: Proceedings of IJCAI. pp. 1683–1688.

[98] Navigli, R., Ponzetto, S. P., 2010. BabelNet: Building a very large multilingual semantic network. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL), Uppsala, Sweden. pp. 216–225.

[99] Navigli, R., Ponzetto, S. P., 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial Intelligence 193, 217–250.

[100] Neelakantan, A., Shankar, J., Passos, A., McCallum, A., 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In: Proceedings of EMNLP. Doha, Qatar, pp. 1059–1069.

[101] Neely, J. H., Keefe, D. E., Ross, K. L., 1989. Semantic priming in the lexical decision task: Roles of prospective prime-generated expectancies and retrospective semantic matching. pp. 1003–1019.

[102] Niemann, E., Gurevych, I., 2011. The people's web meets linguistic knowledge: automatic sense alignment of Wikipedia and WordNet. In: Proceedings of the Ninth International Conference on Computational Semantics. pp. 205–214.

[103] Palmer, M., Dang, H., Fellbaum, C., 2007. Making fine-grained and coarse-grained sense distinctions, both manually and automatically. Natural Language Engineering 13 (2), 137–163.

[104] Pantel, P., Lin, D., 2002. Discovering word senses from text. In: Proceedings of KDD. pp. 613–619.

[105] Pennacchiotti, M., De Cao, D., Basili, R., Croce, D., Roth, M., 2008. Automatic induction of FrameNet lexical units. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. EMNLP '08. pp. 457–465.

[106] Pham, N. T., Lazaridou, A., Baroni, M., 2015. A multitask objective to inject lexical contrast into distributional semantics. In: Proceedings of ACL. pp. 21–26.

[107] Pilehvar, M. T., Jurgens, D., Navigli, R., 2013. Align, Disambiguate and Walk: a Unified Approach for Measuring Semantic Similarity. In: Proceedings of ACL. pp. 1341–1351.

[108] Pilehvar, M. T., Navigli, R., 2014. A large-scale pseudoword-based evaluation framework for state-of-the-art Word Sense Disambiguation. Computational Linguistics 40 (4), 837–881.

[109] Pilehvar, M. T., Navigli, R., 2014. A robust approach to aligning heterogeneous lexical resources. In: Proceedings of ACL. pp. 468–478.

[110] Pilehvar, M. T., Navigli, R., 2015. From senses to texts: An all-in-one graph-based approach for measuring semantic similarity. Artificial Intelligence 228, 95–128.

[111] Ponzetto, S. P., Strube, M., 2007. Knowledge derived from Wikipedia for computing semantic relatedness. Journal of Artificial Intelligence Research (JAIR) 30, 181–212.

[112] Pradhan, S., Loper, E., Dligach, D., Palmer, M., 2007. SemEval-2007 task-17: English lexical sample, SRL and all words. In: Proceedings of SemEval. pp. 87–92.

[113] Proisl, T., Evert, S., Greiner, P., Kabashi, B., 2014. Semantiklue: Robust semantic similarity at multiple levels using maximum weight matching. SemEval 2014, 532–540.

[114] Radinsky, K., Agichtein, E., Gabrilovich, E., Markovitch, S., 2011. A word at a time: Computing word relatedness using temporal semantic analysis. In: Proceedings of the 20th International Conference on World Wide Web. WWW '11. pp. 337–346.

[115] Reisinger, J., Mooney, R. J., 2010. Multi-prototype vector-space models of word meaning. In: Proceedings of ACL. pp. 109–117.

[116] Resnik, P., 1995. Using information content to evaluate semantic similarity in a taxonomy. In: Proceedings of IJCAI. pp. 448–453.

[117] Rothe, S., Schütze, H., July 2015. 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). Association for Computational Linguistics, Beijing, China, pp. 1793–1803.

[118] Rubenstein, H., Goodenough, J. B., 1965. Contextual correlates of synonymy. Communications of the ACM 8 (10), 627–633.

[119] Ruiz, P., Poibeau, T., 2015. El92: Entity linking combining open source annotators via weighted voting. In: 9th International Workshop on Semantic Evaluation (SemEval 2015). pp. 355–359.

[120] Salton, G., Wong, A., Yang, C. S., 1975. A vector space model for automatic indexing. Communications of the ACM 18 (11), 613–620.

[121] Schütze, H., Pedersen, J., 1995. Information retrieval based on word senses. In: Proceedings of SDAIR'95. Las Vegas, Nevada, pp. 161–175.

[122] Schwartz, R., Reichart, R., Rappoport, A., 2015. Symmetric pattern based word embeddings for improved word similarity prediction. CoNLL 2015, 258–267.

[123] Sinha, R., Mihalcea, R., 2007. Unsupervised graph-based Word Sense Disambiguation using measures of word semantic similarity. In: Proceedings of ICSC. pp. 363–369.

[124] Snow, R., O'Connor, B., Jurafsky, D., Ng, A., 2008. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In: Proc. of EMNLP-08. pp. 254–263.

[125] Snow, R., Prakash, S., Jurafsky, D., Ng, A. Y., 2007. Learning to merge word senses. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). Prague, Czech Republic, pp. 1005–1014.

[126] Søgaard, A., Agić, Ž., Alonso, H. M., Plank, B., Bohnet, B., Johannsen, A., 2015. Inverted indexing for cross-lingual NLP. In: The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2015). pp. 1713–1722.

[127] Strube, M., Ponzetto, S. P., 2006. WikiRelate! Computing Semantic Relatedness Using Wikipedia. In: Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2. AAAI'06. Boston, Massachusetts, pp. 1419–1424.

[128] Tufiş, D., Ion, R., Bozianu, L., Ceauşu, A., Ştefănescu, D., 2008. Romanian wordnet: Current state, new applications and prospects. In: Proceedings of 4th Global WordNet Conference, GWC. pp. 441–452.

[129] Turney, P. D., Littman, M. L., Bigham, J., Shnayder, V., 2003. Combining independent modules to solve multiple-choice synonym and analogy problems. In: Proceedings of Recent Advances in Natural Language Processing. Borovets, Bulgaria, pp. 482–489.

[130] Turney, P. D., Pantel, P., 2010. From frequency to meaning: Vector space models of semantics. Journal of Artificial Intelligence Research 37, 141–188.

[131] Tversky, A., Gati, I., 1982. Similarity, separability, and the triangle inequality. Psychological Review 89 (2), 123–154.

[132] Vannella, D., Jurgens, D., Scarfini, D., Toscani, D., Navigli, R., 2014. Validating and Extending Semantic Knowledge Bases using Video Games with a Purpose. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Baltimore, USA, pp. 1294–1304.

[133] Vasilescu, F., Langlais, P., Lapalme, G., 2004. Evaluating Variants of the Lesk Approach for Disambiguating Words. In: LREC.

[134] Venhuizen, J. N., Basile, V., Evang, K., Bos, J., 2013. Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Short Papers. Ch. Gamification for Word Sense Labeling, pp. 397–403.

[135] Vickrey, D., Biewald, L., Teyssier, M., Koller, D., 2005. Word sense disambiguation for machine translation. In: Proceedings of Conference on Empirical Methods in Natural Language Processing. Vancouver, Canada, pp. 771–778.

[136] Webber, W., Moffat, A., Zobel, J., 2010. A similarity measure for indefinite rankings. ACM Transactions on Information Systems (TOIS) 28 (4), 1–38.

[137] Weissenborn, D., Hennig, L., Xu, F., Uszkoreit, H., 2015. Multi-Objective Optimization for the Joint Disambiguation of Nouns and Named Entities. 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). Beijing, China, pp. 596–605.

[138] Weissenborn, D., Xu, F., Uszkoreit, H., 2015. Dfki: Multi-objective optimization for the joint disambiguation of entities and nouns & deep verb sense disambiguation. 9th International Workshop on Semantic Evaluation (SemEval 2015), 335–339.

[139] Weston, J., Bordes, A., Yakhnenko, O., Usunier, N., 2013. Connecting language and knowledge bases with embedding models for relation extraction. In: Proceedings of EMNLP. Seattle, Washington, USA, pp. 1366–1371.

[140] Wu, Z., Palmer, M., 1994. Verbs semantics and lexical selection. In: Proceedings of the 32^{nd} Annual Meeting on Association for Computational Linguistics. ACL '94. Las Cruces, New Mexico, pp. 133–138.

[141] Xu, J., Croft, W. B., 1996. Query expansion using local and global document analysis. In: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '96. pp. 4–11.

[142] Yeh, E., Ramage, D., Manning, C. D., Agirre, E., Soroa, A., 2009. WikiWalk: random walks on Wikipedia for semantic relatedness. In: Proceedings of the Workshop on Graph-based Methods for Natural Language Processing. pp. 41–49.

[143] Zhong, Z., Ng, H. T., 2010. It makes sense: A wide-coverage Word Sense Disambiguation system for free text. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL). Uppsala, Sweden, pp. 78–83.

[144] Zhong, Z., Ng, H. T., 2010. It Makes Sense: A wide-coverage Word Sense Disambiguation system for free text. In: Proceedings of the ACL System Demonstrations. pp. 78–83.