Artificial Intelligence 228 (2015) 95–128

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

From senses to texts: An all-in-one graph-based approach for measuring semantic similarity

Mohammad Taher Pilehvar *, Roberto Navigli

Department of Computer Science, Sapienza University of Rome, Italy

ARTICLE INFO

Article history:
Received 3 September 2014
Received in revised form 30 June 2015
Accepted 9 July 2015
Available online 15 July 2015

Keywords:
Semantic similarity
Lexical semantics
Semantic Textual Similarity
Personalized PageRank
WordNet graph
Semantic networks
Word similarity
Coarsening WordNet sense inventory

ABSTRACT

Quantifying semantic similarity between linguistic items lies at the core of many applications in Natural Language Processing and Artificial Intelligence. It has therefore received a considerable amount of research interest, which in its turn has led to a wide range of approaches for measuring semantic similarity. However, these measures are usually limited to handling specific types of linguistic item, e.g., single word senses or entire sentences. Hence, for a downstream application to handle various types of input, multiple measures of semantic similarity are needed, measures that often use different internal representations or have different output scales. In this article we present a unified graph-based approach for measuring semantic similarity which enables effective comparison of linguistic items at multiple levels, from word senses to full texts. Our method first leverages the structural properties of a semantic network in order to model arbitrary linguistic items through a unified probabilistic representation, and then compares the linguistic items in terms of their representations. We report state-of-the-art performance on multiple datasets pertaining to three different levels: senses, words, and texts.

1. Introduction

The measurement of semantic similarity is an essential component of many applications in Natural Language Processing (NLP) and Artificial Intelligence (AI). Measuring the semantic similarity of text pairs enables the evaluation of the output quality of machine translation systems [1] or the recognition of paraphrases [2], while laying the foundations for other fields, such as textual entailment [3,4], information retrieval [5,6], question answering [7,8], and text summarization [9]. At the word level, semantic similarity can have direct benefits for areas such as lexical substitution [10] or simplification [11], and query expansion [12], whereas, at the sense level, the measurement of semantic similarity of concept pairs can be utilized as a core component in many other applications, such as reducing the granularity of lexicons [13,14], Word Sense Disambiguation [15], knowledge enrichment [16], or alignment and integration of different lexical resources [17–20].

As a direct consequence of their design, most of the current approaches to semantic similarity are limited to operating at specific linguistic levels. For instance, similarity approaches for large pieces of texts, such as documents, usually utilize the statistics obtained from the input items [21–24] and, therefore, are inapplicable for pairs of linguistic items with small contextual information, such as words or phrases. A unified approach that can enable the efficient comparison of linguistic items at different linguistic levels would be able to free downstream NLP applications from needing to consider the type of

* Corresponding author.
E-mail addresses: pilehvar@di.uniroma1.it (M.T. Pilehvar), navigli@di.uniroma1.it (R. Navigli).

http://dx.doi.org/10.1016/j.artint.2015.07.005
0004-3702/

items being compared. However, despite the potential advantages, very few approaches have attempted to cover different linguistic levels: most previous work has focused on tuning or extending existing approaches to other linguistic levels, rather than proposing a unified similarity measurement method. For instance, sense-level measures have been extended to the word level by assuming the similarity of word pairs as being that of the closest senses of the two words [25], whereas word-level approaches have been utilized for measuring the similarity of text pairs [26]. However, these approaches do not usually work on the extended levels as effectively as they do on the original ones. For instance, measures for concept semantic similarity often fall far behind the state of the art when extended for use in measuring the similarity of word pairs [27–29].

In this article, we propose a unified approach to semantic similarity that can handle items from multiple linguistic levels, from sense to text pairs. The approach brings together two main advantages: (1) it provides a unified representation for all linguistic items, enabling the meaningful comparison of arbitrary items, irrespective of their scales or the linguistic levels they belong to (e.g., the phrase *take a walk* to the verb *stroll*); (2) it disambiguates linguistic items to a set of intended concepts prior to modeling and, hence, it is able to identify the semantic similarities that exist at the deepest sense level, independently of the text's surface forms or any semantic ambiguity therein. For example, consider the following two pairs of sentences:

a1. *Officers fired.*
a2. *Several policemen terminated in corruption probe.*

b1. *Officers fired.*
b2. *Many injured during the police shooting incident.*

Surface-based approaches that are merely based on string similarity cannot capture the similarity between any of the above pairs of sentences as there exists no lexical overlap. In addition, a surface-based semantic similarity approach considers both *a1* and *b1* as being identical sentences, whereas we know that different meanings of the verb *fire* are triggered in the two contexts.

In our recent work [30] we presented *Align, Disambiguate, and Walk* (ADW), a graph-based approach for measuring semantic similarity that can overcome both these deficiencies: firstly, it transforms words to senses prior to modeling, hence providing a deeper measure of similarity comparison and, secondly, it performs disambiguation by taking into account the context of the paired linguistic item, enabling the same linguistic item to have different meanings when paired with different linguistic items. Our technique models arbitrary linguistic items through a unified representation, called *semantic signature*, which is a probability distribution over concepts, word senses, or words in a lexicon. Thanks to this unified representation, our approach can compute the similarity of linguistic items at and across arbitrary levels, from word senses to texts. We also proposed a novel approach for comparing semantic signatures which provided improvements over the conventional cosine measure. Our approach for measuring semantic similarity obtained state-of-the-art performance on several datasets pertaining to different linguistic levels. In this article, we extend that work as follows:

1. we propose two novel approaches for injecting out-of-vocabulary (OOV) words into the semantic signatures, obtaining a considerable improvement on datasets involving many OOV entries while calculating text-level semantic similarity;
2. we provide an approach for creating a semantic network from Wiktionary and show that it can be used effectively for generating semantic signatures and for comparing pairs of items;
3. we re-design experiments in the sense and text levels in order to have a more meaningful comparison of different similarity measurement techniques and also perform evaluation on more datasets at the word level.

The rest of this article is organized as follows. We first introduce in Section 2 the three main linguistic levels upon which we focus in this article. We then provide an overview of the related work in Section 3. Section 4 explains how we constructed different semantic networks to be used as underlying resources of our approach. A detailed description of our similarity measurement approach, i.e., ADW, is provided in Section 5, followed by our experiments for evaluating the proposed technique at different linguistic levels in Section 6. Finally, we provide the concluding remarks in Section 7.

2. Semantic similarity at different levels

Measuring the semantic similarity of pairs of linguistic items can be performed at different linguistic levels. In this work, we focus on three main levels: senses, words, and sentences. Table 1 lists example semantic similarity judgments for pairs belonging to each of these three linguistic levels.[1] In our example in Table 1(a), which is based on the WordNet 3.0 sense inventory [31], the precious stone sense of the noun *jewel* (jewel^1_n) is paired with three senses of the noun *gem*: *gem*^5_n, which is synonymous to *jewel*^1_n being the stone used in jewelry, *gem*^3_n, which refers to a brilliant and precious person, and *gem*^4_n, which is synonymous to *muffin* that is a sweet baked bread. In the sense-level similarity, the task is to compute the

[1] Following [15], we denote the ith sense of the word w with the part of speech p as w^i_p in the reference inventory.

Table 1
Example semantic similarities of pairs of items pertaining to three different linguistic levels: senses, words, and sentences. Sense numbers and definitions are from WordNet 3.0.

Source sense	$jewel_n^1$: a precious or semiprecious stone incorporated into a piece of jewelry
Similarity level	Target sense
High	gem_n^5: *synonymous to* $jewel_n^1$ *according to WordNet 3.0*
Medium	gem_n^3: a person who is as brilliant and precious as a piece of jewelry
Low	gem_n^4: a sweet quick bread baked in a cup-shaped pan

(a) Sense level.

Source word	jewel
Similarity level	Target word
High	precious_stone
Medium	gold
Low	paper

(b) Word level.

Source sentence	Human influence has been detected in warming of the atmosphere and the ocean
Similarity level	Target sentence
High	There has been evidence that humans caused global warming
Medium	In many ways, industrialization is negatively impacting our world today
Low	Earth's tilt is the reason behind the existence of different seasons

(c) Sentence level.

degree of semantic similarity of a pair of concepts. This similarity is high for $jewel_n^1$ and gem_n^5 which are both referring to the same jewelry object. The third sense of *gem*, though still related to the jewelry sense, has a lower degree of similarity with $jewel_n^1$ and gem_n^5 as it is a metaphor referring to a person who possesses the qualities of a jewel. On the other hand, the bread sense of *gem* has no semantic similarity to $jewel_n^1$ (nor does it have to any of the two other above-mentioned senses of *gem*). Note that the sense-level similarity measurement can also involve the similarity of different senses of the same word, e.g., to perform sense clustering [13,14].

At the word level, semantic similarity usually corresponds to the similarity of the closest senses of the paired words [25]. In our word-level example in Table 1(b), *jewel* is shown as having a high similarity to *precious stone*, owing to their overlapping meaning, i.e., "a precious or semiprecious stone incorporated into a piece of jewelry." For the medium similarity example, we pair *jewel* with *gold* as both are common elements of jewelries. The terms *jewel* and *paper* do not have any senses in close connection with one another.

At the sentence level, similarity judgment should ideally indicate the amount of information shared between a pair of sentences. In the case of our sentence-level example in Table 1(c), the sentence with high similarity preserves important concepts and semantics of the source sentence, i.e., global warming and the influence of humans on it. Instead, the sentence with the medium similarity does not share the global warming information with the source sentence and mentions the slightly different concept of the negative impact of industrialization on our world today. The unrelated sentence in the low similarity example has almost no overlapping concept with the source sentence.

From the above examples we observe that the same task of semantic similarity involves an inherently different focus as we move from one linguistic level to another: the similarity for the case of senses and words is characterized by the direct similarity of the concepts they surrogate, while for larger textual items similarity denotes the amount of overlapping information between the two items. As a result of this inherent difference, similarity measurement approaches have usually focused on a single linguistic level only. In the following Section we provide the related work on each of the three aforementioned levels.

3. Related work

We review the techniques for semantic similarity measurement according to the linguistic level they can be applied to: sense, word, and text level, and summarize the most important approaches at each level.

3.1. Sense-level similarity

Sense-level measures for semantic similarity are mostly based on lexical resources. These measures have often viewed lexical resources as semantic networks and then used the structural properties of these networks in order to compute semantic similarity. As a *de facto* community standard lexicon, WordNet has played an important role for computing semantic similarity between concepts. A series of WordNet-based measures directly exploit the structural properties of WordNet, such as path length and depth in the hierarchy [32–36], whereas others utilize additional information from external corpora to overcome the associated problems, such as varying link distances in lexicons [37–39]. A comprehensive survey of

WordNet-based measures is provided in [25]. Other WordNet-based measures rely less on the structure of this resource. Instead they take into account overlaps in sense definitions [40,41], or leverage monosemous words in the definitions to create web search queries and hence by this means gather representative contextual information for a given concept. The topic signatures method presented in [42] is an instance of the latter category, which represents each sense as a vector over corpus-derived features. However, topic signatures rely heavily on the availability of representative monosemous words in the definitions, and this lowers their coverage [43]. In contrast, our approach provides a rich and high-coverage representation of WordNet senses, irrespective of their frequency, and it outperforms all other sense similarity approaches in different sense clustering experiments.

In addition to WordNet, other lexical resources have also been used for measuring concept-to-concept semantic similarity. Several approaches have exploited information from dictionaries such as the Longman Dictionary of Contemporary English, thesauri such as Roget's [44] and Macquarie [45], or integrated knowledge resources such as BabelNet [18] for their similarity computation [46–50]. Collaboratively-constructed resources such as Wikipedia [51–53] have also been used extensively as lexical resources for measuring semantic similarity.

There have also been efforts to use distributional techniques for modeling individual word senses or concepts. The main hypothesis in the distributional approaches is that similar words appear in similar contexts, where context can include windows of surrounding words or semantic relationships [54,55]. The sense-specific distributional models usually carry out clustering on a word's context and obtain "multi-prototype" vectors that are sensitive to varying contexts, i.e., vectors that can represent individual senses of words [56–58]. However, the sense representations obtained are usually not linked to any sense inventory, a linking that thus has to be carried out either manually, or with the help of sense-annotated data. Chen et al. [59] addressed this issue by exploiting word sense definitions in WordNet and applied the obtained representations to the task of Word Sense Disambiguation. SensEmbed [60] is another recent approach that obtains sense-specific representations by employing neural network based learning on large amounts of sense-annotated texts. Similarly to the other above-mentioned corpus-based approaches, SensEmbed is prone to the coverage issue as it can only learn representations for those senses that are covered in the underlying corpus.

3.2. Word-level similarity

Among the different levels, the word level is the one that has attracted the greatest attention over the past decade, with several datasets dedicated to the evaluation of word similarity measurement [61–63]. The approaches at this level can be grouped into two categories: distributional and lexical resource-based. Distributional models [64] are the prevailing paradigm for modeling individual words [24], and they lie at the core of several similarity measurement techniques [65]. This paradigm aims at modeling a word on the basis of the context in which it usually appears. The conventional distributional techniques use cooccurrence statistics for the computation of vector-based representations of different words [21,66]. The earlier models in this branch [21,67] take as a word's context only its bag of surrounding words, while more sophisticated contexts such as grammatical dependencies [68,69] or selectional preferences on the argument positions [70], have also been considered. The weights in cooccurrence-based vectors are usually computed by means of tf–idf [71] or Pointwise Mutual Information [72,73], and the dimensionality of the resulting weights matrix is often reduced, for instance using Singular Value Decomposition [74–76]. Topic models [77,78] are another suite of techniques that model a word as a probability distribution over a set of topics. The structured textual content of specific lexical resources such as the encyclopedic Wikipedia has also been used for distributional word similarity [27,79].

A recent branch of distributional models uses neural networks to directly learn the expected context of a given word and model it as a continuous vector [80,81], often referred to as word embedding. Representation of words as continuous vectors, however, has a long history [82,83]. The resurgence of these models stemmed from the work of Bengio et al. [84], who introduced a Multilayer Perceptron (MLP) designed for statistical language modeling. Two prominent contributions in this path were later made through the works of Collobert and Weston [85], who extended and applied the model to several NLP applications, and Mikolov et al. [86], who simplified the original model, providing significant reduction in training time.

Lexical resource-based approaches usually make an assumption that the similarity of two words can be calculated in terms of the similarity of their closest senses, hence enabling any sense-level measure to be directly applicable for comparing word pairs. One can use this method, as we did, for evaluating several WordNet-based sense-level measures on standard word-level benchmarks [25], such as RG-65 [61] and MC-30 [87] datasets. Recently, larger collaborative resources such as Wikipedia and Wiktionary have also been leveraged for measuring word similarity [88–92]. Most similar to our approach are random walk-based methods that model words through the stationary distributions of the Personalized PageRank algorithm on the WordNet graph [93,94], the Wikipedia graph [89], or other graphs obtained from dependency-parsed text [95]. However, unlike our approach, none of these techniques disambiguates the words being compared, and they hence consider a word as a conflation of all its meanings, which potentially reduces the quality of similarity measurement. We show the benefit arising from the disambiguation phase in our word-level experiments.

3.3. Text-level similarity

Text-level methods can be grouped into two categories: (1) those that view a text as a combination of words and calculate the similarity of two texts by aggregating the similarities of word pairs across the two texts, and (2) those that model a text as a whole and calculate the similarity of two texts by comparing the two models obtained. Approaches in the first category search for pairs of words across the two texts that maximize similarity and compute the overall similarity by aggregating individual similarity values, either by exploiting large text corpora [96,26,97–99], or thesauri [100], sometimes also taking into account the word order in the text [101].

The second category usually involves transforming texts into vectors and computing the similarity of texts by comparing their corresponding vectors. Vector space models [21] are an early example of this category, an idea borrowed from Information Retrieval. The initial models mainly focused on the representation of larger pieces of text, such as documents, where a text is modeled on the basis of the frequency statistics of the words it contains. Such models, however, suffer from sparseness and cannot capture similarities between short text pairs that use different wordings. A more suitable vector representation for shorter textual items is one that is based on semantic composition, and that seeks to model a text by combining the representations of its individual words [102]. A thorough study and comparison of different compositionality strategies is provided in [103–105]. Recently, an approach mixing distributional information and explicit knowledge has been successfully applied to cross-lingual document retrieval and categorization [106].

Despite the fact that they totally ignore semantics, string-based similarity techniques which treat texts as sequences of characters have shown themselves to be strong baselines for measuring semantic similarity [29,107,108]. The Longest Common Subsequence [109] and Greedy String Tiling [110] are examples of such string-based measures. Among other measures, that fall into the second category and are closest to our approach, are random walk-based approaches [111,89], which also function at the word level and were described above. These methods, however, do not involve a sense disambiguation step and therefore potentially suffer from ambiguity, particularly in the case of shorter textual items. In contrast, our approach has the advantage of providing explicit disambiguation for the compared linguistic items as a byproduct of the similarity measurement.

4. Preliminaries: semantic networks

The sole, and yet fundamental, resource upon which our semantic similarity measurement algorithm, ADW, relies is a semantic network. A semantic network is a graph structure for representing knowledge. Each node in this graph represents an entity, such as a word or a concept, and edges are semantic relations that link related entities to each other. Any network with such properties can be used in our algorithm. In this article, we consider four semantic networks with different properties: two semantic networks obtained from manually-crafted lexical resources and two automatically-induced ones. As for our manually-constructed lexical resources, we opted for WordNet [31], which is the *de facto* community standard sense inventory, and Wiktionary,[2] which is a collaboratively-constructed online dictionary. The nodes in the WordNet semantic network represent individual concepts, while edges denote manually-crafted concept-to-concept relations. Wiktionary provides a larger word coverage in comparison to WordNet, thanks to its collaborative nature. However, the resource is not readily representable as a semantic network. In what follows, we first briefly describe how we build our two WordNet-based and Wiktionary-based networks (Sections 4.1 and 4.2, respectively), and then explain our procedure for the automatic construction of two semantic networks by leveraging distributional semantic models (Section 4.3).

4.1. WordNet 3.0

Synsets are the basic building blocks of WordNet. Each synset represents a distinct concept that is lexicalized by a group of synonymous words (e.g., {*airplane*, *aeroplane*, *plane*} is the synset for the fixed-wing aircraft concept). Synsets in WordNet are connected to each other by means of semantic and lexical relations. Therefore, WordNet can be viewed as a semantic network of interconnected concepts, where each node in the network is a synset (i.e., concept) and edges are modeled after the relations encoded in WordNet.

In Fig. 1 we illustrate a small subgraph of WordNet, partly showing the neighborhood of the synset containing the fixed-wing aircraft sense of the noun *airplane* (the node at the center of the figure). As shown in the figure, edges in the WordNet graph are typed, with the main types being hypernymy[3] and meronymy.[4] However, we do not utilize these types in our semantic graph and consider an edge as a undirected relation between two synsets. The WordNet graph was further enriched in our experiments by connecting a sense with all the other senses that appear in its disambiguated gloss, as given by the Princeton Annotated Gloss Corpus.[5] The corpus is an attempt at disambiguating the content words in WordNet's glosses, providing a suitable means of improving the connectivity among synsets in the WordNet network. For instance, consider the definition for the above-mentioned sense of *airplane*:

[2] http://www.wiktionary.org.

[3] *X* is a hypernym of *Y* if *X* is the generalization *Y* (e.g., *amphibian* is the hypernym of *frog*).

[4] *X* is a meronym of *Y* if *X* is a part or a member of *Y* (e.g., *wheel* is a meronym of *car*).

[5] http://wordnet.princeton.edu/glosstag.shtml.

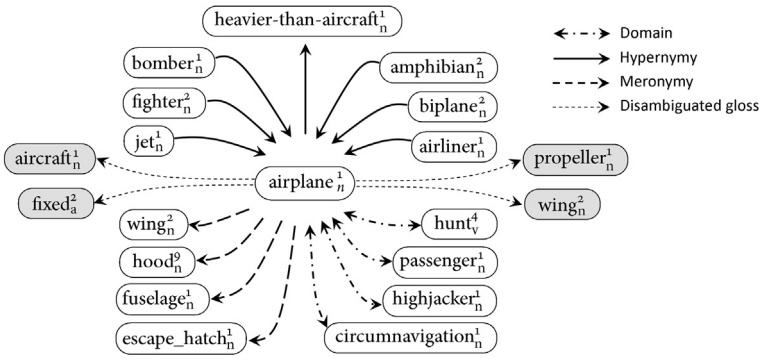

Fig. 1. A small subgraph of WordNet, partly showing the neighborhood of the first sense of the noun *airplane*, i.e., $airplane^1_n$ (one sense per synset is shown in the figure). Nodes in grey are those that are connected to $airplane^1_n$ as a result of the graph enrichment process performed with the help of the disambiguated glosses.

$airplane^1_n$ – an aircraft that has a fixed wing and is powered by propellers or jets

which is disambiguated in the Annotated Gloss Corpus as follows:

$airplane^1_n$ – an $aircraft^1_n$ that has a $fixed^2_a$ $wing^2_n$ and is powered by $propellers^1_n$ or jets.

In our enriched WordNet graph the corresponding synset of the above-mentioned sense of *airplane* is directly linked to the synsets containing the disambiguated senses of the content words in its definition: $aircraft_n$, $fixed_a$, $wing_n$ and $propeller_n$. These nodes are highlighted in grey in Fig. 1. Note that not all the content words are provided with their intended senses in the disambiguated gloss (e.g., jet_n in our example). The resulting WordNet graph contains about 118K nodes (synsets) covering around 155K unique terms. The average node degree is 8.9 in this undirected graph.

4.2. Wiktionary

Recent years have seen the surge of a wide range of AI applications that exploit the vast amount of semi-structured knowledge available in collaboratively-constructed resources [112]. High coverage is the main feature of these resources, thanks to their massive number of contributors. Since the coverage of ADW directly depends on the number of words covered by the underlying semantic network, a natural extension of our approach would be to utilize a high-coverage semantic network for the generation of semantic signatures. In order to verify the applicability of our approach to such resources, we picked a large collaboratively-constructed online dictionary, i.e., Wiktionary. Thanks to its collaborative nature, Wiktionary provides a larger and more up-to-date vocabulary than WordNet. For instance, this resource has a wider coverage of named entities (e.g., *Ferrari*), multi-word expressions (e.g., *full_ throttle*), alternative forms or spellings (e.g., *Mejico*), abbreviations and acronyms (e.g., *adj.* and *MIME*), slang terms (e.g., *homewrecker*), and domain-specific words (e.g., *data_type*). However, similarly to many other machine-readable dictionaries, Wiktionary is not readily representable as a semantic network. Therefore, the resource has first to be transformed into a semantic network before it can be utilized by ADW for the generation of semantic signatures. There are two main obstacles to the generation of a rich semantic network of Wiktionary, both of which are due to the difficult nature of connecting lexicographic senses by means of semantic relations:

- *Sparse set of relations*: Unlike WordNet, Wiktionary does not have the benefit of a rich set of semantic relations. In fact, to our estimate, less than 20% of word entries in Wiktionary are provided with at least one lexical semantic relation (e.g., synonymy and hypernymy). As a result, attempts at exploiting these pre-defined relations for the transformation of Wiktionary into an ontology generally fail, as they can only produce sparse semantic networks with many of the nodes left in isolation [113].
- *Lack of sense-disambiguated information*: Wiktionary relations are not disambiguated on the target side. For instance, consider the first sense of the noun *windmill* in Wiktionary, which is defined as "A machine which translates linear motion of wind to rotational motion by means of adjustable vanes called sails." Wiktionary lists the noun *machine* as the hypernym of this sense. However, there is no mechanism to make the intended sense of *machine* distinguishable from the others. Hence, a linkage between that sense of *windmill* and the intended sense of *machine* carries sense information on the source side only, as the intended sense is only known for *windmill* and not for *machine*. As a result, the relations provided by Wiktionary first need to be disambiguated according to its sense inventory, before they can be used to connect different word senses to each other.

We presented in [43] an approach that addresses both issues and transforms an arbitrary machine-readable dictionary into a full-fledged semantic network. The gist of the approach lies in its collection of related words from the definition of a

word sense. These words are subsequently disambiguated using a similarity-based disambiguation technique, resulting in a set of links between pairs of word senses.

Specifically, we first create an empty undirected graph $G = (S, E)$ such that S is the set of word senses in Wiktionary and $E = \emptyset$. For each source word sense $s \in S$ we gather a set of related words $W = \{w_1, \ldots, w_n\}$, which comprises all the hyperlinked words in the definition of s and, if available, additional relations from Wiktionary (e.g., synonyms). If the word w_i is monosemous according to Wiktionary's sense inventory, the procedure is trivial and an edge is introduced in G between s and the only sense of w_i. However, if w_i is polysemous, we need to disambiguate the target side of the edge, i.e., the related word w_i. To this end, we measure the similarity between the definition of s and the definitions of all the senses of w_i. To measure this similarity, we opt for ADW when using the WordNet graph (more details in Section 5). The sense of w_i that produces the maximal similarity with s is taken as the intended sense $\hat{s}_{w_i}$ of w_i, and the edge $(s, \hat{s}_{w_i})$ is accordingly added to E. In this procedure we make the assumption that the most important content words in the Wiktionary definitions are usually provided with hyperlinks.

We illustrate the graph construction procedure by way of an example. Consider the Wiktionary page for the noun *windmill* in Fig. 2 (left). The goal is to identify, for each sense of this noun, a set of related word senses. In the figure we have highlighted the hyperlinked words in the definitions by rectangles, and underlined the pre-defined Wiktionary relations. Consider the first sense:

> $windmill_n^1$: A *machine* which translates *linear* motion of *wind* to *rotational* motion by means of *adjustable vanes* called *sails*,

in which seven hyperlinked terms are shown in italics. Consider the noun *machine* in this definition. The word is disambiguated by computing the semantic similarity between the context in which it appears, i.e., the definition of $windmill_n^1$, and all the definitions of *machine* in Wiktionary (right side in Fig. 2). The sense of *machine* which produces the maximal similarity is taken as the intended sense (shown by a star sign in the figure) and accordingly an edge is introduced into the graph between this sense and $windmill_n^1$.

All the highlighted words, i.e., the hyperlinked words in the definitions and the pre-defined Wiktionary relations, are disambiguated using the same similarity-based disambiguation procedure. As a result of performing this procedure for our example word sense $windmill_n^1$, new edges are added to the graph connecting this sense to the following related word senses: $machine_n^1$, $linear_a^6$, $wind_n^1$, $rotational_a^1$, $adjustable_a^1$, $vane_n^2$, and $sail_n^5$.[6] Based on the described procedure, we constructed a semantic network of Wiktionary containing around 372K nodes, each denoting a word sense with any of the four open-class parts of speech: noun, verb, adjective, and adverb.[7] Our Wiktionary graph has more than three times the number of nodes in the WordNet 3.0 graph. The average node degree in this undirected graph is around 4.4.

We further enrich the Wiktionary graph by exploiting the multilingual knowledge available in this resource. Our approach utilizes translations of words in other languages as bridges between synonymous words in English, a technique that is usually used in paraphrasing [114]. Specifically, we first obtain all the translations for each sense s of word w in Wiktionary. Assume that the sense s of w translates to the word t_l in language l. We hypothesize that an English word sense s' of w' is synonymous or closely related to s, if it is also translated into t_l in language l. Hence, we introduce an edge between these two senses s and s' in the graph. In order to avoid ambiguity, as t_l we only consider words that are monosemous according to the Wiktionary sense inventory for language l. For instance, the Finnish noun *ammatti*, which is monosemous according to Wiktionary, links six English word senses: $career_n^1$, $business_n^2$, $occupation_n^1$, $trade_n^6$, $calling_n^2$, and $vocation_n^2$.[8] This procedure results in about 500 additional nodes and more than 35K new edges, increasing the average node degree by 0.1. We refer to this Wiktionary graph as *WKT* in our experiments.

We also constructed a variant of the Wiktionary semantic network in which, in addition to the hyperlinked words that were used in the *WKT* graph, the set of related words W for a word sense also includes the non-hyperlinked content words in the definition. This graph, called *WKTall*, has 429K nodes with an average degree of 10. In Section 6.3.3 we report the results of the evaluations carried out on ADW when using this variant of the Wiktionary graph.

4.3. Automatically-induced semantic networks

Directly connected entities in a semantic network are expected to share most of the semantics, i.e., to be the most semantically related ones. Therefore, having at hand a procedure for computing the most semantically related entities to

[6] $machine_n^1$: "A device that directs and controls energy, often in the form of movement or electricity, to produce a certain effect.", $linear_a^6$: "A type of length measurement involving only one spatial dimension.", $wind_n^1$: "Real or perceived movement of atmospheric air usually caused by convection or differences in air pressure.", $rotational_a^1$: "Of, pertaining to or caused by rotation.", $adjustable_a^1$: "capable of being adjusted", $vane_n^2$: "Any of several usually relatively thin, rigid, flat, or sometimes curved surfaces radially mounted along an axis, as a blade in a turbine or a sail on a windmill, that is turned by or used to turn a fluid.", and $sail_n^5$: "The blade of a windmill".

[7] In our experiments we used the Wiktionary version `20131002`, which provides definitions for around 447K word senses.

[8] $career_n^1$: "One's calling in life; a person's occupation; one's profession"; $business_n^2$: "A person's occupation, work, or trade."; $occupation_n^1$: "An activity or task with which one occupies oneself; usually specifically the productive activity, service, trade, or craft for which one is regularly paid; a job."; $trade_n^6$: "The skilled practice of a practical occupation"; $calling_n^2$: "A job or occupation"; $vocation_n^2$: "An occupation for which a person is suited, trained or qualified".

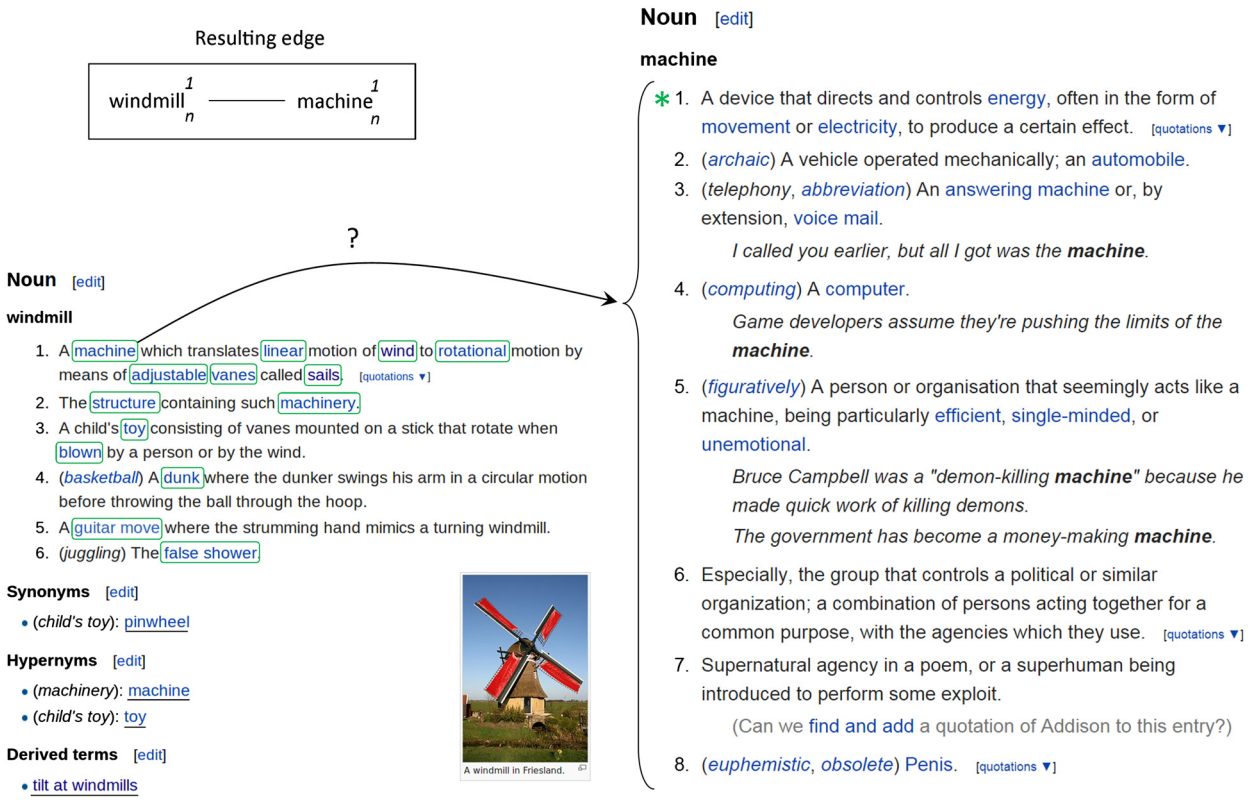

Fig. 2. A snapshot of the Wiktionary page for the noun *windmill* (left). The hyperlinked words in the definitions are highlighted in rectangles and the pre-defined Wiktionary relations are underlined. The similarity-based disambiguation procedure automatically disambiguates the noun *machine* in the definition of the first sense of windmill, i.e., $windmill_n^1$, to the first of its eight senses in the same sense inventory (right). The disambiguation is the outcome of the fact that the definition of $windmill_n^1$ (which contains the target word *machine*) produces maximal similarity with the definition of $machine_n^1$. As a result of this disambiguation, an edge is introduced into the graph between $windmill_n^1$ and $machine_n^1$.

a given entity, one can think of automatically constructing a semantic network. A popular technique for modeling the semantics of linguistic items is the distributional hypothesis, according to which semantically similar items are expected to appear in similar contexts. Baroni and Lenci [115] provide an overview of the distributional semantic models (DSM). In order to evaluate the suitability of automatically-induced semantic networks for the construction of semantic signatures, we used two different DSM techniques for the construction of semantic networks: a conventional frequency-based vector space model and a state-of-the-art continuous model based on deep neural networks.

However, in order to be able to utilize DSM techniques to automatically induce sense-based semantic networks, i.e., graphs whose nodes are word senses or concepts, large sense-annotated corpora are required (see [43] for a pseudoword-based solution). Due to the lack of such corpora, we are limited to the construction of word-based semantic networks, i.e., graphs with words as their nodes, unlike WordNet and Wiktionary networks whose nodes represent concepts and word senses, respectively. Most similar to our computation of semantic similarity on automatically-induced networks is the work of Iosif and Potamianos [116], which exploits cooccurrence statistics for the construction of semantic networks and then exploits the structural information of the networks obtained for the computation of semantic similarity. In Section 6.3.4 we present our experiments on utilizing the automatically-induced semantic networks in the task of word similarity measurement.

4.3.1. Distributional thesaurus (DM)

Conventional DSMs take as context any word appearing in the vicinity of a target word, irrespective of the syntactic or semantic relation between the two [74,24]. Structured models improve this by encoding the relationship between a word and its context, hence providing a richer and more sophisticated model of meaning. Baroni and Lenci [115] provide an overview of structured DSMs, models in which the context words are limited to only those that are linked by a syntactic relation or lexical pattern. TypeDM is a structured DSM in which third-order tensors, i.e., ternary geometrical objects that model distributional data in terms of word–link–word tuples, are calculated in such a way as to assign more importance to relations that tend to take more forms [115,117]. Baroni and Lenci released a set of TypeDM vectors estimated by means of Local Mutual Information (LMI) on a 2.8 billion-token corpus obtained by concatenating the ukWaC corpus, English

Table 2
The 10 entries most related to the three words *smartphone*$_n$, *paper*$_n$, and *terminate*$_v$ in the TypeDM-based distributional thesaurus used for the generation of a semantic network.

smartphone$_n$	paper$_n$	terminate$_v$
handheld$_n$	report$_n$	renegotiate$_v$
handset$_n$	article$_n$	renew$_v$
PC$_n$	book$_n$	sign$_v$
laptop$_n$	document$_n$	cancel$_v$
iphone$_n$	pamphlet$_n$	suspend$_v$
ipod$_n$	booklet$_n$	void$_v$
i-mode$_n$	text$_n$	negotiate$_v$
console$_n$	newspaper$_n$	stop$_v$
next-generation$_n$	newsletter$_n$	rescind$_v$
workstation$_n$	essay$_n$	end$_v$

Table 3
The ten most related entries to the three words *smartphone*, *paper*, and *terminate* according to the pre-trained Word2vec vectors on the Google News dataset (about 100 billion words).

smartphone	paper	terminate
smartphones	They_unroll_toilet	terminated
handset	papers	terminating
Android_smartphone	Accelerating_3G	termination
smart_phones	cloth_swabs	terminates
Android_phones	quoted_Hao_Peng	unilaterally_terminate
Android	newspaper	rescind
Android_smartphones	newsprint_uncoated	discontinue
netbook	printed	suspend
Android_OS	Qassas_confirmed	cancel
touchscreen_smartphone	8_#/#-by-##-inch	revoke

Wikipedia, and the British National Corpus [81].[9] Based on this model, Partha Pratim Talukdar constructed a distributional thesaurus (see footnote 9). The thesaurus lists the top ten nearest neighbors of each word in a vocabulary of about 31K nouns, verbs, and adjectives, calculated using the cosine distance between TypeDM tensors. Table 2 shows the neighbors of three words *smartphone*$_n$, *paper*$_n$, and *terminate*$_v$ in this thesaurus. We transform the TypeDM thesaurus into a semantic network and use the resulting network in our experiments for the generation of semantic signatures. The graph, called DM hereafter, comprises 30.7K nodes belonging to three parts of speech, nouns (20K), verbs (5K), and adjectives (5K), which are linked to each other by means of around 250K undirected edges.

4.3.2. Word embeddings (W2V)

The past few years have seen a resurgence of interest in the usage of neural networks for processing massive amounts of texts. Continuous vector representations, also known as word embeddings, are a prominent example [84,118,86,58]. In this representation, the vectors' weights are directly computed so as to maximize the probability of the context in which the word being modeled tends to appear. This permits efficient representation of models trained on massive amounts of data in relatively small-sized vectors. We used the 300-dimensional vectors trained on the 100 billion-word Google News dataset provided as a part of the Word2vec toolkit.[10] The model covers more than 3 million words and phrases, which is a considerable vocabulary size. For each entry, we computed the ten most similar entries using the scripts provided in the toolkit. Table 3 shows the top ten closest words to our three example words *smartphone*, *paper*, and *terminate*. Accordingly, we construct the W2V semantic network by restricting the entries to those containing at least one alphanumeric character, including also apostrophe, period, hyphen and underscore. The resulting graph has around 2.9M nodes and an average node degree of 17.

5. A unified semantic representation

So far we have described how we construct our semantic networks. In this Section we proceed by explaining how these networks are used for the measurement of semantic similarity. Fig. 3 illustrates the process of measuring the semantic similarity of a pair of linguistic items using our similarity measurement technique. Our approach, ADW, consists of two main steps: an **A**lignment-based **D**isambiguation of the two linguistic items and a random **W**alk on a semantic network in

[9] http://clic.cimec.unitn.it/dm/.
[10] http://code.google.com/p/word2vec/.

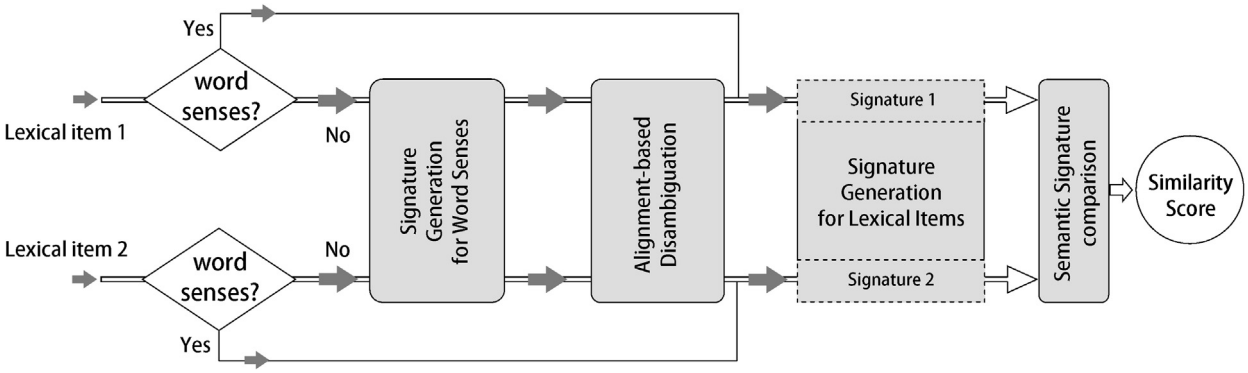

Fig. 3. The process of measuring the semantic similarity of a pair of linguistic items using our approach, ADW. A linguistic item is first disambiguated into a set of concepts, if not already sense disambiguated, after which its semantic signature is computed. The similarity of two linguistic items is then calculated by comparing their semantic signatures.

order to obtain and compare their semantic representations. We term our representation for a given linguistic item as its *semantic signature*. Our approach for the generation of semantic signatures is a graph-based one that models a linguistic item as a probability distribution over all entities in a lexicon. The weights in this distribution denote the relevance of the corresponding entity to the modeled linguistic item.

We start this section by providing, in Section 5.1, a formal description of how we leverage random walks on semantic networks in order to model arbitrary linguistic items through semantic signatures. We then present, in Section 5.2, four methods (one of which is proposed by us) for comparing the semantic signatures obtained and calculating the similarity score for two linguistic items. Finally, in Section 5.3, we explain how the semantic signatures of concepts enable our alignment-based disambiguation of a pair of lexical items.

5.1. Semantic signature of a lexical item

Generally speaking, a semantic signature can be viewed as a special form of vector space model (VSM) representation [24]. Similarly to the VSM representation of a linguistic item, the weight associated with a dimension in a semantic signature denotes the relevance or importance of that dimension for the linguistic item. The main difference, however, is in the way the weights are calculated. In a VSM representation, each dimension usually corresponds to a separate word whose weight is often computed on the basis of cooccurrence statistics, whereas in a semantic signature a linguistic item is represented as a probability distribution over all entities in a semantic network where the weights are estimated on the basis of structural properties of the network. For the generation of our semantic signatures, we use the Personalized PageRank (PPR) algorithm [119]. In what follows, we briefly describe the PageRank algorithm and its personalized version.

5.1.1. PageRank

The PageRank algorithm [120] is a celebrated graph analysis technique which can be used to estimate the structural importance of nodes in a graph. PageRank is the best-known algorithm used by Google for ranking different web pages in its search engine results. PageRank represents the web as a graph and estimates the importance of a web page on the basis of the structural properties of the graph. The algorithm has been successfully used in various fields including NLP where it has found numerous applications: sentiment polarity detection [121], Word Sense Disambiguation [122–125], semantic similarity [93,94,89], keyword extraction [126], and lexical resource alignment [127,20].

A simple way to describe the PageRank algorithm is to consider a user who surfs the web by randomly clicking on hyperlinks. The probability that the surfer will click on a specific hyperlink is given by the PageRank value of the page to which the hyperlink points. According to the PageRank algorithm, this probability for a given page is calculated on the basis of the number of its incoming links and their importance. The basic idea is that the more links there are from more important pages, the higher the PageRank value is. This is based on the assumption that important websites are linked to by many other pages. The original PageRank also assumes that the surfer will get bored after a finite number of clicks and will jump to some other page at random. The probability that our surfer will continue surfing by clicking on the hyperlinks is given by a fixed-value parameter, usually referred to as the *damping factor*.

In the original PageRank algorithm the graph models the web with web pages as nodes and hyperlinks between web pages as directed edges. In our formulation, the underlying graph is a semantic network with nodes representing concepts and edges acting as the semantic relationships between concepts. Formally, the PageRank algorithm first represents a semantic network consisting of N concepts as a row-stochastic transition matrix $\mathbf{M} \in \mathbb{R}^{N \times N}$. For instance, consider the graph

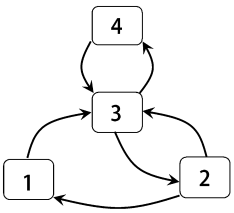

Fig. 4. A graph with 4 nodes and 6 directed edges.

in Fig. 4 that has 4 nodes and 6 directed edges. The graph can be represented as a Markov chain $\mathbf{M}$ where the cell M_{ij} is set to $outDegree(i)^{-1}$ if there exists a link from i to j, and to zero otherwise[11]:

$$\mathbf{M} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1/2 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

Each row in the matrix is a stochastic vector. Note that we calculated the probability of following any outlink as $outDegree(i)^{-1}$, which assumes that all the links are equally likely to be selected. This assumption can be replaced by any other weighting scheme that guarantees a row-stochastic matrix. For instance, one can assign a higher probability to a certain node based on a priori knowledge available. Irrespective of the procedure used for the construction of the matrix $\mathbf{M}$, the PageRank values are given by the principal left eigenvector $\mathcal{S}$ of the matrix:

$$\mathcal{S}\,\mathbf{M} = \lambda\,\mathcal{S} \tag{1}$$

where the eigenvalue λ is one, hence the principal eigenvector. The ith value of the vector $\mathcal{S}$ denotes the PageRank value for the ith page. Different methods have been proposed for the computation of the PageRank values [128]. One popular approach is the power iteration method. According to this iterative method, the PageRank vector $\mathcal{S}$ can be calculated as:

$$\mathcal{S}^{t+1} = (1-\alpha)\,\mathcal{S}^0 + \alpha\,\mathbf{M}\,\mathcal{S}^t \tag{2}$$

where $\mathcal{S}^0$ is a column vector of size N in which the probability mass is distributed among all dimensions, i.e., each cell is assigned a value equal to $\frac{1}{N}$. The α parameter is the damping factor, which is usually set to 0.85 [120]. The procedure is repeated for a fixed number of iterations or until the following convergence criterion is fulfilled:

$$|\mathcal{S}^{t+1} - \mathcal{S}^t| < \epsilon \tag{3}$$

where ϵ is set to 0.0001 in our experiments. The power method finds only one maximal eigenvalue together with its corresponding eigenvector. The resulting eigenvector is a vector of size N with non-negative values. The PageRank algorithm takes the eigenvector as a stationary probability distribution that contains the PageRank values for all the nodes in the graph [128].

5.1.2. Personalized PageRank

The Personalized PageRank (PPR) algorithm is a variation of the PageRank algorithm in which the computation is biased so as to obtain weights denoting the importance with respect to a particular set of nodes. When the graph is a semantic network in which edges denote semantic relationships, this importance can be viewed as the degree of semantic relatedness. Therefore, the PPR algorithm can essentially be used on a semantic network in order to calculate the semantic relatedness of all concepts to a specific concept or set of concepts.

According to the random surfer explanation, there is an additional assumption in the PPR formulation that when the random surfer gets bored, (s)he does not pick a page from the set of all pages in the web, but from a specific set of *personalized* pages. Therefore, in this variant of the PageRank algorithm random restarts are always initiated from a set of specific personalized web pages.

The essential difference in the calculation of the PPR values is in the initialization of the $\mathcal{S}^0$ vector. Instead of distributing the probability mass among all dimensions, the personalization vector $\mathcal{S}^0$ is constructed by concentrating the probability mass on a subset of dimensions only. Hence, the PPR algorithm can be used to obtain a semantic signature for a set of m concepts C. To this end, it is enough to uniformly distribute the probability mass in the personalization vector $\mathcal{S}^0$ among all the corresponding dimensions of C, i.e., each dimension is assigned a probability equal to $\frac{1}{m}$. The resulting semantic

[11] Outdegree is the number of edges starting from a given node in a directed graph.

Table 4
Top-8 dimensions in the semantic signatures generated on the WordNet semantic network for the first sense of *plant* (industrial plant) on the left and the phrase *linux operating system* on the right. For each dimension (shown by some of the word senses of the corresponding synset), we show the associated weight in the initialization vector ($\mathcal{S}^0$), weight computed after one iteration ($\mathcal{S}^1$), together with the final weight ($\mathcal{S}^f$).

$plant_n^1$				*Linux operating system*			
$\mathcal{S}^0$	$\mathcal{S}^1$	$\mathcal{S}^f$	Dimension (synset)	$\mathcal{S}^0$	$\mathcal{S}^1$	$\mathcal{S}^f$	Dimension (synset)
1.000	0.150	0.172	industrial_plant$_n^1$, plant$_n^1$, works$_n^2$	0.500	0.075	0.089	operating_system$_n^1$, os$_n^3$
0.000	0.000	0.009	factory$_n^1$, manufactory$_n^1$	0.500	0.075	0.085	linux$_n^1$
0.000	0.000	0.008	building$_n^1$, edifice$_n^1$	0.000	0.000	0.022	trademark$_n^2$
0.000	0.000	0.007	industrial$_a^1$	0.000	0.000	0.021	unix$_n^1$, unix_operating_system$_n^1$, unix_system$_n^1$
0.000	0.000	0.006	carry_on$_v^1$, conduct$_v^1$, deal$_v^{10}$	0.000	0.000	0.016	konqueror$_n^1$
0.000	0.000	0.006	refinery$_n^1$	0.000	0.000	0.016	edition$_n^1$, variant$_n^1$, variation$_n^4$, version$_n^2$
0.000	0.000	0.006	communication_equipment$_n^1$	0.000	0.000	0.015	open-source$_a^1$
0.000	0.000	0.006	labor$_n^2$, labour$_n^4$, toil$_n^1$	0.000	0.000	0.012	computer_program$_n^1$, program$_n^7$, programme$_n^4$

signature denotes the semantic relevance of each node in the network with respect to the personalized concepts C. This signature can also be computed as the average of the signatures obtained for the individual entities in C (cf. Appendix A for the proposition proof). This makes it possible to calculate the signatures for all nodes in a graph in advance, and later use the pre-computed signatures for the calculation of the semantic signature of an arbitrary linguistic item without needing to re-run the PPR algorithm for that specific item. In our experiments, we used the UKB[12] off-the-shelf implementation of the algorithm.

Example Table 4 shows the top-8 dimensions in the semantic signatures generated on the WordNet semantic network for: (1) $plant_n^1$: the first sense of the noun *plant* in WordNet 3.0 (industrial plant) and, (2) the phrase *linux operating system*. For each dimension (represented by some of the word senses in its corresponding synset) we show the associated weight in the initialization vector ($\mathcal{S}^0$), the weight computed after the first iteration of the algorithm ($\mathcal{S}^1$), and the final weight ($\mathcal{S}^f$). In the case of $plant_n^1$, the semantic signature is obtained by putting all the probability mass in the personalization vector $\mathcal{S}^0$ on the dimension corresponding to that specific sense (i.e., all values in the distribution are set to zero except the one corresponding to the synset containing $plant_n^1$, which is set to one). Instead, for the phrase *linux operating system*, the probability mass in $\mathcal{S}^0$ is distributed among all dimensions corresponding to all senses of all the content words, i.e., *linux* and *operating system*. Since both these are monosemous according to the WordNet sense inventory, the weight in the personalization vector $\mathcal{S}^0$ for our phrase is concentrated on the dimensions corresponding to their only senses, i.e., $linux_n^1$ and $operating\ system_n^1$, and the other dimensions are set to zero. As can be seen from the table, upon the first iteration of the PageRank algorithm (column $\mathcal{S}^1$), none of the uninitialized dimensions are assigned a weight greater than or equal to 0.001, even those that are directly connected to the initialized nodes (e.g., $factory_n^1$ for $plant_n^1$ and $trademark_n^2$ for *linux operating system*). For the case of both examples the highest-ranking dimensions in the final vectors ($\mathcal{S}^f$) correspond to synsets (concepts) that are closely related to the modeled linguistic items. Also, note that the top-ranking synsets do not necessarily belong to the same part of speech. For instance, $industrial_a^1$ and $open\text{-}source_a^1$ are adjectival word senses that are strongly related to our nominal linguistic items $plant_n^1$ and *linux operating system*, respectively.

5.2. Semantic signature similarity

Once we have obtained the semantic signature representations for a pair of linguistic items, we can calculate the similarity of the two items by comparing their corresponding semantic signatures. We adopt four techniques for comparing our semantic signatures: two methods that have been used extensively in previous work on comparing vectors, i.e., Jensen–Shannon divergence and cosine, and two rank-based comparison metrics, i.e., Rank-Biased Overlap and Weighted Overlap.

- **Jensen–Shannon divergence.** This measure is based on the Kullback–Leibler divergence, which is commonly referred to as KL divergence, and is computed for a pair of semantic signatures (probability distributions in general) $\mathcal{S}_1$ and $\mathcal{S}_2$ as:

$$D_{KL}\left(\mathcal{S}_1 \| \mathcal{S}_2\right) = \sum_{h \in H} log_e \left(\frac{\mathcal{S}_1^h}{\mathcal{S}_2^h}\right) \mathcal{S}_1^h \tag{4}$$

  where $\mathcal{S}^h$ is the weight assigned to the dimension h in the semantic signature $\mathcal{S}$ and H is the set of overlapping dimensions across the two signatures. However, KL divergence is non-symmetric. Therefore, we use in our experiments Jensen–Shannon (JS) divergence, which is a symmetrized and smoothed version of KL divergence:

$$D_{JS}\left(\mathcal{S}_1, \mathcal{S}_2\right) = \frac{1}{2} D_{KL}\left(\mathcal{S}_1 \,\middle\|\, \frac{\mathcal{S}_1 + \mathcal{S}_2}{2}\right) + \frac{1}{2} D_{KL}\left(\mathcal{S}_2 \,\middle\|\, \frac{\mathcal{S}_1 + \mathcal{S}_2}{2}\right) \tag{5}$$

[12] http://ixa2.si.ehu.es/ukb/.

- **Cosine.** The measure computes the similarity of two multinomial distributions $\mathcal{S}_1$ and $\mathcal{S}_2$ by treating each as a vector and then computing the normalized dot product of the two signatures' vectors:

$$Sim_{Cos}(\mathcal{S}_1, \mathcal{S}_2) = \frac{\mathcal{S}_1 \cdot \mathcal{S}_2}{\|\mathcal{S}_1\| \, \|\mathcal{S}_2\|} \tag{6}$$

The above-mentioned measures are all calculated by directly incorporating the actual weights in the vectors. There is another class of measures which rely rather on the relative rankings of the entities in the vectors. The most prominent example of this type of statistical measure is the Spearman rank correlation or Spearman's ρ, which computes the statistical dependence between two ranked variables. However, the Spearman correlation does not provide a suitable basis for comparing semantic signatures. The reason behind this drawback is that the measure places as much importance on differences in the ranks of the top elements in the signatures as it does on the ones at the bottom; however, we know that the top elements in the semantic signatures are the most representative ones. Therefore, a suitable measure has to penalize the differences among the top ranks more than it does for the bottom ones. Webber et al. [129] referred to this property as the top-weightedness of a measure. Kendell's τ [130] is another rank-based measure that does not satisfy this property. A thorough overview of different rank similarity methods is provided in [129]. Webber et al. [129] also proposed a top-weighted rank similarity measure, called Rank-Biased Overlap, and evaluated it on the task of comparing search engine results and assessing retrieval systems.

- **Rank-Biased Overlap (RBO).** Let H_d be the set of overlapping dimensions between the top-d elements of the two signatures $\mathcal{S}_1$ and $\mathcal{S}_2$. The RBO measure is then calculated as:

$$RBO(\mathcal{S}_1, \mathcal{S}_2) = (1-p) \sum_{d=1}^{|H|} p^{d-1} \frac{|H_d|}{d} \tag{7}$$

where $|H|$ is the number of overlapping dimensions between the two signatures and $p \in [0, 1]$ is a parameter that determines the relative importance of the top elements: smaller p values result in higher top-weightedness. In our experiments we set p to the high value of 0.995, as suggested in [129] for large vectors.

- **Weighted Overlap.** We also employ a fourth measure, Weighted Overlap (WO), that we introduced in [30]. The measure computes the similarity between a pair of ranked lists by comparing the relative rankings of the dimensions. Let H denote the intersection of all non-zero dimensions in the two signatures and $r_h(\mathcal{S})$ be a function returning the rank of the dimension h in the sorted signature $\mathcal{S}$. Then WO calculates the similarity of two signatures $\mathcal{S}_1$ and $\mathcal{S}_2$ as:

$$Sim_{WO}(\mathcal{S}_1, \mathcal{S}_2) = \frac{\sum_{h \in H} \left(r_h(\mathcal{S}_1) + r_h(\mathcal{S}_2)\right)^{-1}}{\sum_{i=1}^{|H|} (2i)^{-1}} \tag{8}$$

where the denominator is a normalization factor that guarantees a maximum value of one. The measure first sorts the two signatures according to their values and then harmonically weights the overlaps between them. The minimum value is zero and occurs when there is no overlap between the two signatures, i.e., $|H| = 0$. The measure is symmetric and satisfies the top-weightedness property, i.e., it penalizes the differences in the higher rankings more than it does for the lower ones.[13] Note that $r_h(\mathcal{S})$ is the rank of the dimension h in the original vector $\mathcal{S}$ and not that in the corresponding vector truncated to the overlapping dimensions H. In our setting, we experiment with the untruncated semantic signatures and all our signatures are equally-sized (the size being equal to the number of nodes in the network). Hence, in our experiments any pair of signatures has identical dimensions, i.e., their intersection has a size equal to that of either of the two signatures. One advantage of WO over RBO is that it does not need any parameter to be set prior to calculation.

5.3. Alignment-based disambiguation

Measures for computing text semantic similarity often operate at the word surface level. However, ideally, each word in a text has first to be analyzed and disambiguated into its intended sense, and then the whole text modeled once it contains only disambiguated words. Moreover, comparison at the surface level can be especially problematic in the case of shorter textual items, such as word or phrase pairs, as there is not enough contextual information to allow an implicit disambiguation of content words' meanings when a combined representation such as VSM is constructed. Our similarity measure, instead, provides a deeper modeling of linguistic items at the sense level. To this end, we propose an alignment-based Word Sense Disambiguation technique that leverages concepts' semantic signatures to disambiguate the content words in a linguistic item. The reason why we did not choose a conventional Word Sense Disambiguation approach was that they are

[13] When talking about rankings, by *higher* we mean ranks that are closer to the top, i.e., the first-ranked element has the highest rank.

Algorithm 1 Alignment-based sense disambiguation.

Input: T_1 and T_2, the sets of word types being compared
Output: P, the set of disambiguated senses for T_1
1: $P \leftarrow \emptyset$
2: **for each** token $t_i \in T_1$
3: $max\_similarity \leftarrow 0$
4: $best\_sense_i \leftarrow null$
5: **for each** token $t_j \in T_2$
6: **for each** $sense_i \in \text{Senses}(t_i)$, $sense_j \in \text{Senses}(t_j)$
7: $similarity \leftarrow \mathcal{R}(sense_i, sense_j)$
8: **if** $similarity > max\_similarity$ **then**
9: $max\_similarity \leftarrow similarity$
10: $best\_sense_i \leftarrow sense_i$
11: **if** $best\_sense_i <> null$ **then**
12: $P \leftarrow P \cup \{best\_sense_i\}$
13: **return** P

generally ineffective for disambiguating short texts, due to lack of sufficient contextual information (consider, for instance, a single-word context). In addition, our alignment-based disambiguation was designed in accordance with psychological studies which suggest that, when making similarity judgments between linguistic items, humans actually perform a pairwise disambiguation of textual items, and that this process results in a biased comparison that favors the similarities between the two sides rather than the differences [131,132]. For instance, consider the word pair *cushion-pillow*, which is assigned the close-to-identical similarity score of 3.84 (in the scale 0 to 4) in the RG-65 dataset [61]. The noun *cushion* is polysemous and has three senses according to the WordNet sense inventory: (1) "a mechanical damper; absorbs energy of sudden impulses", (2) "the layer of air that supports a hovercraft or similar vehicle", and (3) "a soft bag filled with air or a mass of padding such as feathers or foam rubber". When *cushion* is paired with the noun *pillow*, its "soft bag" sense is triggered (*pillow* is a hyponym of the "soft bag" *cushion* in WordNet), resulting in a high similarity score, assigned by several human annotators.

We show the procedure for our alignment-based disambiguation in Algorithm 1. The algorithm takes as its input the sets T_1 and T_2 of word types on the two comparison sides. For a word $t_i \in T_1$, the algorithm searches for a sense $best\_sense_i$ that produces the maximal similarity with a specific sense among all the senses of all the words in T_2. This procedure is repeated for all words in T_1 and finally, as output, the set P of disambiguated senses for word types in T_1 is returned. In line 6, Senses(t) returns all senses of the word t and $\mathcal{R}(sense_1, sense_2)$, on the next line, measures the similarity of $sense_1$ and $sense_2$ by leveraging our semantic similarity technique at the sense level. At this level, the similarity of a pair of senses is computed by generating their semantic signatures and comparing the senses' signatures with the help of any of the comparison methods described in Section 5.2. Note that in our disambiguation procedure we assume one sense per discourse [133] and, in case multiple instances of a word exist in a linguistic item, all are assigned the same sense. We explain the disambiguation procedure using our two example sentences from Section 1:

a1. Officers fired.
a2. Several policemen terminated in corruption probe.

We are interested in disambiguating the content words in both sentences: $officer_n$ and $fire_v$ in *a1* and $policeman_n$, $terminate_v$, $corruption_n$, and $probe_n$ in *a2*. As we show in Fig. 5, among all possible pairings of all the senses of $fire_v$ to all the senses of all words in *a2*, the sense $fire_v^4$ (the employment termination sense) obtains the maximal similarity value (to $terminate_v^4$ with $\mathcal{R}(fire_v^4, terminate_v^4) = 1$), and hence it is selected as the sense for $fire_v$ in sentence *a1*. Fig. 6 illustrates the final, maximally-similar sense alignment of the word types in *a1* and *a2*. The source side in each alignment is taken as the intended sense of its corresponding word (shaded in grey in Fig. 6). Note that the procedure has to be repeated in the other direction in order to disambiguate word types in *a2*. For example, $probe_n$ is disambiguated to its first sense (defined as "an inquiry into unfamiliar or questionable activities") after being aligned to the second sense of $officer_n$ (defined as "someone who is appointed or elected to an office and who holds a position of trust"). On the other hand, $officer_n$ is disambiguated to its third sense (defined as "a member of a police force") after being aligned to its synonym $policeman_n^1$ in the other sentence. The resulting alignment produces the following sets of disambiguated senses for the two pairs of sentences from Section 1:

$P_{a1} = \{officer_n^3, fire_v^4\}$
$P_{a2} = \{policeman_n^1, terminate_v^4, corruption_n^6, probe_n^1\}$

$P_{b1} = \{officer_n^3, fire_v^2\}$
$P_{b2} = \{injure_v^2, police_n^1, shooting_n^1, incident_n^2\}$

where P_x denotes the corresponding set of senses of sentence x. We note that since the textual items can have different lengths, the alignments are not necessarily one-to-one or symmetrical. Also, given that our automatically-constructed

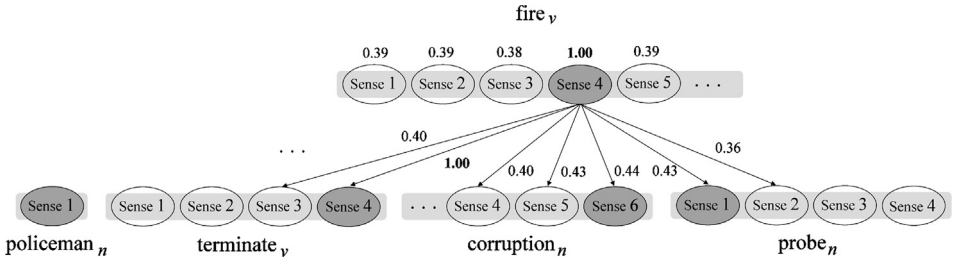

Fig. 5. Potential alignments of the fourth sense of the verb *fire* (in sentence *a1*) to some of the senses of the word types in sentence *a2*, along with their similarity values.

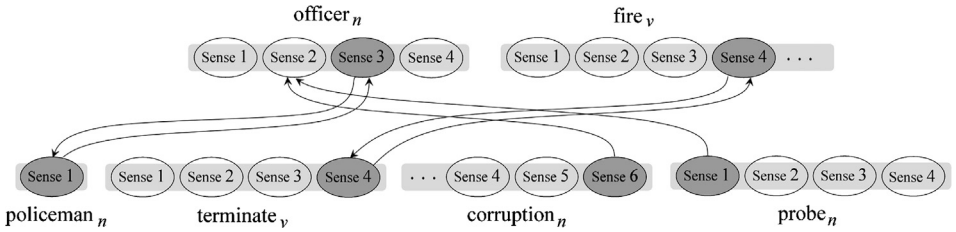

Fig. 6. Alignments which maximize the similarities across words in *a1* and *a2* (the source side of an alignment is taken as the intended sense of its corresponding word).

graphs, i.e., *DM* and *W2V*, do not provide sense distinctions (see Section 4.3), they cannot be used for the alignment-based disambiguation. Therefore, we apply the disambiguation phase only when experimenting with the WordNet and Wiktionary graphs.

We further demonstrate the advantage that our alignment-based disambiguation approach can provide in comparison with the conventional disambiguation techniques by means of examples from two existing standard datasets for two tasks: word similarity and phrase-to-word similarity. In the RG-65 dataset [61], which is a standard evaluation framework for word similarity, the noun *crane* is paired with three other nouns: *rooster*, *bird*, and *implement* with the respective similarity scores of 1.41, 2.68, and 2.37 (in the scale 0–4). A conventional disambiguation technique falls short of disambiguating either word in each pair as single words do not have any context. In contrast, our algorithm disambiguates the noun *crane* into its fifth sense in WordNet 3.0, defined as "large long-necked wading bird of marshes and plains in many parts of the world", when the noun is paired with *rooster* or *bird*. In the context of *implement*, the fourth sense of the noun *crane* is triggered, i.e., $crane_n^4$: "lifts and moves heavy objects; lifting tackle is suspended from a pivoted boom that rotates around a vertical axis." As for the phrase-to-word similarity, consider the following example from the training set of the SemEval-2014 task on Cross-Level Semantic Similarity [134]:

- *leak*
- *sifting through cracks in the ceiling*

Our alignment-based disambiguation identifies $leak_v^3$, defined as "enter or escape as through a hole or crack or fissure", as the intended sense of *leak*. The sense obtains maximal similarity with the first sense of the noun *crack*, defined as "a long narrow opening." A conventional approach is ineffective for disambiguating such cases in which the context is either not conclusive or non-existing.

Disambiguation of long textual items Our disambiguation step is specifically designed for short linguistic items, such as words or short phrases, which do not have enough contextual information. In the case of larger linguistic items such as sentences or paragraphs, the presence of a suitable number of content words guarantees an implicit disambiguation of the terms in the linguistic items. The hunch is that, similarly to VSM techniques such as ESA [27], when the representations of multiple content words are aggregated, a partial disambiguation takes place. As an example, consider the linguistic item "plant for manufacturing household appliances." The noun plant has two main senses, i.e., the living organism and the industrial plant. The semantic signature generated for the sole noun *plant* gives importance to concepts that are relevant to both these senses. However, when *plant* is put together with a word such as *manufacturing* in some linguistic item, the overlapping industrial meanings of the two give rise to the weights of industry-related concepts in the resulting semantic signature. Therefore, in the semantic signature representation of the above linguistic item, concepts related to the industrial sense will have higher weights than those related to the living organism meaning, hence an implicit disambiguation of the noun *plant* already at the lexical level.

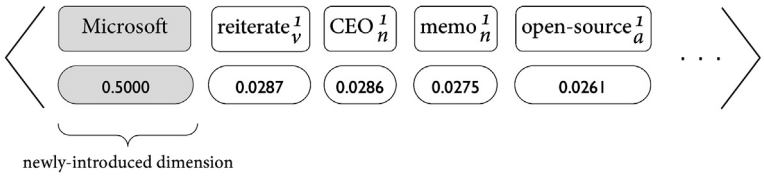

Fig. 7. Direct OOV handling of the OOV word *Microsoft* for the semantic signature generated for the example sentence *h2*. Note that each dimension is a synset that contains the shown word sense and the dimensions are sorted by their associated weights.

5.4. OOV handling

Similarly to any other graph-based approach that maps words in a given textual item to their corresponding nodes in a semantic network, our approach for modeling linguistic items through semantic signatures can suffer from its limited coverage of words: it can handle only those words that are associated with some nodes in the underlying semantic network. As a result, the semantic signature generation phase ignores out-of-vocabulary (OOV) words in a textual item, as they are not defined in the corresponding lexical resource and hence do not have an associated node in the semantic graph for the random walk to be initialized from. This can be particularly problematic when measuring semantic similarity of text pairs that contain many OOV words, such as infrequent named entities, acronyms or jargon. In order to alleviate this issue, we propose two novel techniques for handling OOV terms while measuring the semantic similarity of textual items. These techniques will be described in the following two subsections.

5.4.1. Direct OOV injection

A semantic signature is essentially a vector whose dimensionality is the number of connected nodes in the underlying semantic network. As mentioned earlier, in a linguistic item's semantic signature the weight assigned to each dimension denotes the relevance of the corresponding concept or word sense to the modeled linguistic item. In the PPR algorithm, random restarts are always initiated from nodes that are associated with a linguistic item. Consequently, the corresponding dimensions of these nodes in the resulting semantic signature possess high weights and are among the top elements in the sorted list of concepts or word senses in that signature. However, if a content word in the linguistic item does not have a corresponding node in the semantic network, it will be ignored in the semantic signature generation. For example, consider the following pair of sentences:

h1. *Steve Ballmer* has been vocal in the past warning that Linux is a threat to *Microsoft*.
h2. In the memo, *Microsoft's* CEO reiterated the open-source threat to Windows.

where the WordNet OOV words are highlighted in *italics*. In the semantic signature obtained on the WordNet semantic network for the first sentence, the top-10 dimensions belong to content words such as *Linux*, *Windows*, *trademark*, *warning*, *threat*, and *vocal*. On the other hand, synsets containing different meanings of *reiterate*, *CEO*, *open-source*, *memo*, and *threat* are among the highest weighted dimensions in the second signature, i.e., the signature obtained for *h2*. However, the two terms *Steve Ballmer* and *Microsoft* are absent from the two signatures since neither of the two words are defined in WordNet and hence they do not have a corresponding node in the WordNet graph.

Here, we propose a technique, namely direct OOV injection, for taking into consideration the words that are not covered during semantic signature generation. According to this procedure, we introduce new dimensions in the resulting semantic signature, one for each OOV term, while assigning a weight to the new dimension so as to place it among the top elements in the semantic signature. This OOV word handling technique can be seen as a back-off to the string-based similarity which, as also mentioned in Section 1, provides a strong baseline in many tasks such as sentence similarity.

For the case of our example, we introduce, in each of the two semantic signatures, new dimensions corresponding to their missing terms, i.e., $Steve\_Ballmer_n$ and $Microsoft_n$ for *h1* and $Microsoft_n$ for *h2*. Fig. 7 illustrates our direct OOV handling for the sentence *h2*. We set the associated weights of the newly-introduced dimensions to 0.5 so as to guarantee their placement among the top dimensions in their corresponding signatures. We utilize this approach for handling OOV entries in our text-level experiments (Section 6.4) and show that it can provide considerable performance improvement on datasets containing many OOV entries. Note that, since we use only the Weighted Overlap measure for comparing pairs of signatures in our sentence-level experiments (see Section 6.4), we do not need to normalize the newly-created vectors because the measure considers the relative ranking of the dimensions and hence is insensitive to the modification of the weights as long as it does not alter the order of the other dimensions.

5.4.2. Wikipedia-assisted OOV mapping

The described direct OOV injection technique is a post-processing step that modifies the semantic signatures subsequent to their generation, by including the missing terms as new dimensions in the signature. Here, we propose an alternative approach that directly replaces an OOV entry in the textual item with its most relevant WordNet concept, prior to the generation of the semantic signature. Given its wide coverage of named entities, Wikipedia provides a suitable means for

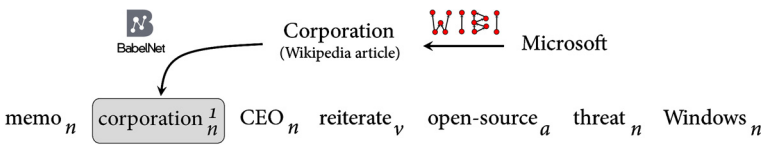

Fig. 8. Wikipedia-assisted OOV handling of the OOV word *Microsoft* in the sentence *h2*. A word's hypernym is obtained from WiBi and mapped to the corresponding WordNet synset using BabelNet.

handling OOV words in WordNet. We therefore leveraged a Wikipedia-derived taxonomy in order to enable the handling of WordNet OOV entries. Given a textual item, the Wikipedia-assisted technique replaces all its WordNet OOV terms with the synsets that best represent them. The replacement is performed in two steps:

1. For a given OOV word w, we first use a Wikipedia taxonomy to obtain the generalization (i.e., hypernym) of w in terms of a Wikipedia article.
2. We then map the obtained hypernym of w to the corresponding synset in WordNet with the help of a mapping of Wikipedia articles to WordNet synsets.

Specifically, as our taxonomy, we considered the Wikipedia Bitaxonomy (WiBi) [135], a state-of-the-art taxonomy derived from Wikipedia. When we encounter an OOV content word w, we use WiBi to extract its hypernym. For instance, for the OOV words *Microsoft* and *Steve Ballmer*, the hypernyms listed are the Wikipedia articles for *corporation* and *businessperson*, respectively. The reason behind our upward move in the taxonomy hierarchy is that, if a word is not defined in WordNet it is probably because it is too domain-specific or not lexicographically relevant. Note that the hypernym of w in WiBi is itself the title of a Wikipedia article. Hence, as a final step, we have to map this article title to the corresponding WordNet synset. For this purpose we utilized BabelNet [18], which is a multilingual encyclopedic dictionary that provides a mapping of Wikipedia pages to WordNet synsets. As a result of this process, if such a mapping exists, the OOV word w is replaced with its WordNet synset in the textual item. In case the word w is ambiguous, i.e., it is associated to multiple word senses in WiBi, we repeat the procedure for all its senses and replace w with all its corresponding WordNet synsets.

For instance, consider our earlier example from Section 5.4.1. After applying this pre-processing stage, the two sentences are transformed into the following:

h1. $\{businessperson_n^1, vocal_a, past_a, warning_n, linux_n, threat_n, corporation_n^1\}$
h2. $\{memo_n, corporation_n^1, ceo_n^1, reiterate_v, open\text{-}source_a, threat_n, windows_n^1\}$

where the OOV words *Microsoft* and *Steve Ballmer* are, respectively, mapped onto the word senses $corporation_n^1$, defined as "a business firm whose articles of incorporation have been approved in some state", and $businessperson_n^1$ defined as: "a capitalist who engages in industrial commercial enterprise" in WordNet 3.0. We show in Fig. 8 our Wikipedia-assisted OOV handling process for the noun *Microsoft* in the sentence *h2*.

During the generation of the semantic signature, the corresponding nodes of the newly-introduced word senses will also be considered as starting points in the random restarts of the PPR algorithm. Therefore, the corresponding node and its neighboring synsets will be assigned higher weights in the resulting semantic signature. We note that, due to the limitations of WiBi and BabelNet, this OOV handling approach can only be applied to the case where the WordNet semantic network is used for the generation of semantic signatures.

6. Experiments

We carried out three sets of experiments in order to evaluate our similarity measurement approach at three different levels: **sense**, **word**, and **sentence** levels. In this section, we first introduce the evaluation benchmarks used in our experiments for the three levels, and this is then followed by a detailed description of the experiments and their corresponding results.

6.1. Benchmarks

We compared the performance of our system with state-of-the-art measures on several benchmarks across three linguistic levels:

- **Sense level**: we performed a binary merging of WordNet senses and evaluated the results on the basis of the sense groupings provided as part of the Senseval-2 English Lexical Sample WSD task, and also the OntoNotes project (Section 6.2).
- **Word level**: we evaluated our system in the judgment correlation task on three standard datasets: RG-65, WordSim-353 similarity, and YP-130 (Section 6.3).

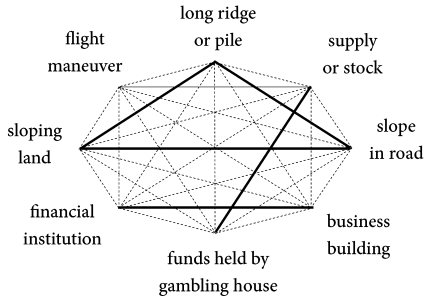

Fig. 9. A sense merging example for eight senses of the noun *bank*. Solid lines indicate sense pairs that are merged according to the Senseval-2 sense groupings.

- **Sentence level**: we took as our test bed the datasets of the SemEval-2012 task on Semantic Textual Similarity (Section 6.4).

As also mentioned in Section 1, previous works have often focused on only one of the above levels and very few have considered multiple tasks or datasets. Hence, for each level and for each dataset, we compare the performance of our approach against the corresponding state-of-the-art results reported in the literature. We describe the comparison systems for each dataset in its corresponding section.

6.2. Experiment 1: sense-level similarity

We start our experiments from the lowest linguistic level, i.e., the sense level. At this level, our approach does not involve the disambiguation phase and hence the procedure reduces to obtaining semantic signatures of each sense and then comparing the senses' semantic signatures by means of a signature comparison method (cf. Section 5.2).

An important application of a sense similarity measurement is that of clustering similar word senses in fine-grained sense inventories. As a case in point, WordNet is believed to have very fine-grained sense distinctions, a feature that is attested to by relatively low levels of annotator agreement in WordNet-based manual sense annotations [136,137]. Reducing the fine granularity of sense inventories can have direct impact on their utility and accordingly on the performance of different NLP applications. For example, a coarsely-defined sense inventory can be more suitable for tasks such as Word Sense Disambiguation [14], information retrieval [138], and Machine Translation [139]. Several earlier works have pursued the task of grouping together similar senses of standard sense inventories such as WordNet [13,14] or other resources such as Wikipedia [140]. We take the task of coarsening the WordNet sense inventory as benchmark for evaluating the performance of our method at the sense level.

6.2.1. Datasets

As our sense clustering datasets, we considered two different resources which provide clustering of WordNet senses:

- The **Senseval-2** English Lexical Sample WSD task [141], which includes sense clusterings for 411 nouns, 519 verbs and 256 adjectives.
- The **OntoNotes** project v4.0 [142], in which 1530 nouns and 1153 verbs have at least two of their senses grouped together.

Following Snow et al. [13] we viewed sense clustering as a binary sense merging problem, which is straightforward to evaluate. To this end, we first converted the sense grouping datasets into sense pairing ones by considering all the possible pairings of senses of each word. Fig. 9 illustrates a sense merging example for eight different senses of the noun *bank* in the Senseval-2 dataset. Highlighted lines in the figure indicate the sense pairs that are grouped together in the same sense cluster in the dataset. Note that the original dataset provides a clustering of different senses of each word as opposed to the binary sense merging shown in the figure.

We show in Table 5, for the two sense pairing datasets, the total number of possible pairings, as well as the proportion of *merged* and *not-merged* pairs for each part of speech. As can be seen from the table, in the Senseval-2 dataset more than a quarter of all adjective sense pairings are set to *merged*, which demonstrates the high degree of fine granularity of adjectives in WordNet. The proportion of merged verb sense pairs in the two datasets is significantly different, 11.3% for Senseval-2 and 26.6% for OntoNotes. This difference is also observed at a lower degree for the noun sense pairs. This suggests that the two datasets provide different levels of granularity reduction for different parts of speech.

6.2.2. Signature comparison methods

In all our sense merging experiments, we evaluate our approach with the four different comparison methods described in Section 5.2:

Table 5
Statistics of the gold-standard sense groupings provided as a part of Senseval-2 English Lexical Sample WSD task and also of OntoNotes v4.0. The words column shows the total number of terms, in the corresponding part of speech, for which there was at least one sense grouping.

Source	POS	Words	Total pairings	Merged
Senseval-2	Nouns	411	15892	15.9%
	Verbs	519	29097	11.3%
	Adjectives	256	6735	27.8%
OntoNotes	Nouns	1530	20665	9.9%
	Verbs	1153	9074	26.6%

- Jensen–Shannon divergence (ADW_{JS}),
- Cosine (ADW_{Cos}),
- RBO (ADW_{RBO}),
- Weighted Overlap (ADW_{WO}).

6.2.3. Semantic graphs

In Section 4, we described how we construct two semantic networks based on two different lexical resources, i.e., WordNet and Wiktionary. However, our two sense grouping datasets are based on the WordNet sense inventory only. This does not allow us to use the *WKT* graph for clustering these datasets. We therefore utilize only the WordNet network in our sense-level experiments.

6.2.4. Comparison systems

We compare the accuracy of our approach in measuring the similarity of sense pairs against seven well-known WordNet-based sense similarity measurement techniques: Lin's universal similarity measure [39, LIN], Jiang and Conrath's combined approach [38, JCN], Wu and Palmer's conceptual similarity [34, WUP], Leacock and Chodorow's Normalized path length [36, LCH], Hirst and St-Onge [35, HSO], Resnik's information-based approach [37, RES], and the Lesk similarity algorithm [40, LESK] as extended by Banerjee and Pedersen [143].

RES, JCN, and LIN measure the similarity of two concepts in a taxonomy based on the notion of information content of the concepts, i.e., "the extent to which they share information in common" [37], whereas WUP, LCH, and HSO use the relative distance of two concepts in the WordNet graph while taking into account the depth of their least common subsumer (WUP), the depth of the hierarchy (LCH), or the number of changes in the direction in the path between two synsets (HSO). LESK takes a different approach and measures the similarity of two senses in terms of the similarity of their definitions in WordNet. A comprehensive survey of these measures is provided in [25].

6.2.5. Experimental setup

We constructed, for each similarity measure, a simple threshold-based binary classifier that sets two senses as *merged* if their semantic similarity is equal to or greater than a certain threshold, and to *not-merged* otherwise. In our experiments we used the WS4J[14] implementation of the above-mentioned WordNet-based measures. All the WordNet sense annotations in both datasets were mapped to senses in version 3.0 so as to allow WS4J to be directly applicable. Also, in order to ensure more reliable estimation of the performance, we performed experiments with 5-fold cross validation. Results are evaluated in terms of precision (P) and recall (R), defined as $P = \frac{tp}{tp+fp}$ and $R = \frac{tp}{tp+fn}$ where *tp*, *fp*, and *fn* denote true positives, false positives, and false negatives, respectively. We also report results for F1, which is the harmonic mean of P and R.

6.2.6. Sense merging results

Tables 6 and 7 list the performance of different variants of our system, as well as the comparison sense similarity measures on the Senseval-2 and OntoNotes datasets, respectively. Results are averaged across five folds and shown separately for each part of speech: nouns, verbs, and adjectives for Senseval-2 and nouns and verbs for the OntoNotes dataset. Note that most of the comparison measures are unable to handle adjectives, and hence we do not report their corresponding performance on the adjectives of the Senseval-2 dataset. We also show in the tables (last row) the performance of a simple baseline system that merges all paired senses, hence attaining an optimal recall of 1.0 but a precision equal to the fraction of pairs to be merged in the corresponding dataset.

As can be seen from Table 6, on the Senseval-2 dataset almost all similarity measures outperform the baseline on nouns and verbs. On adjectives, however, only two of the comparison approaches provide outputs, neither of which can improve over the baseline. On this dataset, among the comparison measures, LCH and HSO provide the best F1 performance while grouping nouns and verbs, respectively. ADW, irrespective of its signature comparison method, yields F1 improvement over the best of the seven comparison techniques. The best overall F1 performance on this dataset is obtained by the Weighted

[14] https://code.google.com/p/ws4j/.

Table 6
Performance of different sense similarity techniques in the binary sense merging of Senseval-2 sense groupings in terms of precision (P), recall (R), and F1 (averaged across five folds). Baseline corresponds to a system that merges all the sense pairings.

Approach	Nouns			Verbs			Adjectives		
	P	R	F1	P	R	F1	P	R	F1
LESK	0.16	1.00	0.27	0.11	1.00	0.20	0.28	1.00	0.43
JCN	0.24	0.35	0.28	0.14	0.71	0.21	–	–	–
LIN	0.27	0.31	0.29	0.24	0.48	0.20	–	–	–
RES	0.29	0.40	0.34	0.29	0.23	0.26	–	–	–
WUP	0.32	0.51	0.39	0.30	0.24	0.27	–	–	–
LCH	0.38	0.43	0.40	0.23	0.24	0.24	–	–	–
HSO	0.44	0.23	0.30	0.31	0.25	0.28	0.28	1.00	0.44
ADW_{JS}	0.41	0.55	0.47	0.47	0.50	0.48	0.41	0.66	0.51
ADW_{Cos}	0.43	0.49	0.46	0.45	0.51	0.47	0.39	0.66	0.49
ADW_{RBO}	0.41	0.55	0.47	0.46	0.49	0.47	0.44	0.64	0.52
ADW_{WO}	0.41	0.61	**0.49**	0.46	0.52	**0.49**	0.45	0.66	**0.53**
Baseline	0.16	1.00	0.27	0.11	1.00	0.20	0.28	1.00	0.44

Table 7
Performance of different sense similarity techniques in the binary sense merging of OntoNotes sense groupings in terms of precision (P), recall (R), and F1 (averaged across five folds). Baseline corresponds to a system that merges all the sense pairings.

Approach	Nouns			Verbs		
	P	R	F1	P	R	F1
JCN	0.10	1.00	0.18	0.27	1.00	0.42
LIN	0.19	0.40	0.26	0.27	1.00	0.42
WUP	0.19	0.39	0.26	0.27	0.91	0.42
RES	0.22	0.33	0.26	0.27	1.00	0.42
LCH	0.23	0.34	0.27	0.27	0.95	0.42
LESK	0.25	0.35	0.28	0.40	0.51	0.45
HSO	0.28	0.25	0.27	0.27	1.00	0.42
ADW_{JS}	0.30	0.52	**0.38**	0.49	0.62	0.55
ADW_{Cos}	0.27	0.55	0.36	0.49	0.62	0.55
ADW_{RBO}	0.28	0.54	0.37	0.48	0.64	0.55
ADW_{WO}	0.30	0.53	**0.38**	0.48	0.68	**0.56**
Baseline	0.10	1.00	0.18	0.27	1.00	0.42

Overlap comparison method, improving the results for the best comparison systems on nouns, verbs, and adjectives by 0.09, 0.21, and 0.09, respectively.

On the OntoNotes dataset, as can be seen from Table 7, most of the systems show improvements over the baseline in the noun sense merging task, whereas only one of the comparison approaches (i.e., LESK) can achieve this on verbs. Similarly to the Senseval-2 dataset, our system attains improvements over the best comparison technique on the OntoNotes dataset (i.e., LESK), irrespective of part of speech or the comparison method.

Among the four signature comparison techniques, Weighted Overlap proves to be the most reliable, providing the highest F1 performance across different parts of speech on both datasets. This demonstrates that our transformation of semantic signatures into ordered lists of concepts and our similarity calculation by rank comparison are helpful.

Discussion As mentioned earlier, previous approaches for measuring similarity between concepts often rely on the path length between two synsets in the WordNet graph or their information content. This strategy, however, fails in many cases when the path between two concepts is either too long or non-existent, or when their lowest superordinate is too general, and therefore has very low information content. Our approach, in contrast, benefits from rich representations of arbitrary linguistic items, irrespective of their size, frequency, or part of speech. To further investigate the advantage of such a representation, we carried out an analysis of the outputs of different similarity measurement methods on the task of sense clustering.

We show in Table 8 the fraction of sense pairs that had to be merged but were judged as completely unrelated (zero similarity score) by different techniques. Statistics are presented for different parts of speech and for both datasets, i.e., Senseval-2 and OntoNotes. We can see from the table that most of the comparison sense similarity approaches cannot handle a large portion of sense pairs and hence judge them as having the minimal similarity value, i.e., zero. RES, LIN, and JCN exploit information content and are sensitive to general superordinates with low information content. HSO allows paths that are restricted to a specific length and to some pre-defined patterns only. Among the tested similarity measures, LCH

Table 8
The fraction of sense pairs that had to be merged but were judged as completely unrelated (zero similarity score) by different sense similarity measurement techniques. We show statistics for different parts of speech and for both datasets: Senseval-2 and OntoNotes.

Approach	OntoNotes		Senseval-2		
	Nouns	Verbs	Nouns	Verbs	Adjectives
JCN	88.9	90.6	92.5	89.5	–
HSO	74.5	78.2	77.2	75.4	95.1
LESK	21.3	33.4	26.2	31.6	73.4
LIN	64.0	86.0	71.9	85.2	–
RES	18.4	79.3	28.5	76.5	–
WUP	0.0	0.0	0.0	0.0	–
LCH	0.0	0.0	0.0	0.0	–
ADW	0.0	0.0	0.0	0.0	0.0

and WUP are the only techniques which provide similarity judgments for all verb and noun sense pairs. These two measures rely only on the path length between two concepts and the depth of their lowest superordinate or the maximum depth of the hierarchy and do not make any assumptions on the lengths or patterns of the paths. This permits them to provide similarity scores for pairs of concept that are distant from each other in the network, or have their lowest superordinate very high in the hierarchy. However, neither of these two measures can be applied to similarity judgment for adjectives. In contrast, thanks to its rich representation of senses, ADW returns more graded similarity values, thereby enabling the effective comparison of any two concepts in the network, irrespective of their parts of speech.

6.3. Experiment 2: word-level similarity

We now proceed from the sense level to the word level. Thanks to its wide range of applications, word similarity has received a considerable amount of research interest, making it one of the most popular tasks in lexical semantics, with numerous evaluation benchmarks and datasets.

6.3.1. Experimental setup

We evaluate the performance of our approach in the similarity judgment correlation framework. Given a list of word pairs, the task is to automatically judge the semantic similarity between each pair. The judgments are ideally expected to be highly correlated with those given by humans. We opted for three different standard word similarity datasets:

- RG-65 [61],
- WordSim-353 Similarity subset (WS-Sim) [63,93],
- YP-130 [62].

The RG-65 dataset was created by Rubenstein and Goodenough [61] to study the relationship between the semantic and contextual similarities of pairs of nouns. The dataset contains 65 word pairs judged by 51 human subjects on a scale of 0 (unrelated) to 4 (synonymy) according to their semantic similarity. The YP-130 dataset was first presented by Yang and Powers [62] as a new benchmark for measuring the semantic similarity of verb pairs. The dataset comprises 130 verb pairs, all of which are single words, partly obtained from the TOEFL [74] and ESL [144] datasets. All the pairs in the dataset were judged by six annotators with a reported average inter-annotator Pearson correlation of 0.866. The WordSim-353 comprises 353 noun pairs created by Lev et al. [63]. However, the similarity scale of the original dataset conflates similarity and relatedness, leading to high similarity scores for pairs such as *computer–keyboard* despite the dissimilarity in their meanings. Agirre et al. [93] separated the pairs in the dataset into two subsets: relatedness and similarity. Given that ADW is targeted at semantic similarity, we opted for the similarity subset of WordSim-353 (WS-Sim) as our evaluation framework. The subset comprises 203 word pairs.

Signature comparison methods We observed in the sense-level experiments that our semantic signature representation can be used effectively for measuring semantic similarity when coupled with any of the four tested measures: cosine, Weighted Overlap, Jensen–Shannon Divergence, and Rank-Biased Overlap. Among the four comparison methods, Weighted Overlap proved to be the most suitable for comparing semantic signatures, by consistently providing the best performance on both datasets and for all parts of speech. Given that the variation among the four measures was observed to be rather small, for brevity, from here on we report results based on the Weighted Overlap comparison method only.

Semantic graphs We report results when our two semantic networks (i.e., WordNet and Wiktionary) were used for the generation of semantic signatures. We also provide a discussion in Section 6.3.3 on the experiments we carried out using the other variant of the Wiktionary graph (*WKTall*), the one which enriched the network with additional content words from the definitions, as well as the two automatically-induced networks from corpus-based semantic models (cf. Section 4).

Table 9
Spearman's ρ and Pearson's r correlation coefficients with human judgments on the RG-65 dataset. Results tagged with † are taken from [148].

Approach	RG-65	
	ρ	r
PMI-SVD [81]	0.74	0.74
ESA [27]†	0.75	0.49
WUP [34]†	0.78	0.80
LCH [145]†	0.79	0.84
HSO [35]†	0.79	0.73
ZG-07 [147]†	0.82	0.49
Agirre et al. [93]	0.83	–
ZMG-08 [90]	0.84	–
Hughes and Ramage [94]	0.84	–
Word2vec [86]	0.84	0.83
ADW_{WN}	0.86	0.81
ADW_{WKT}	**0.92**	**0.91**

Comparison systems We compared the performance of our system against state-of-the-art approaches on the datasets. Our results are benchmarked against the word-level extensions of three sense-level measures: LCH [145], WUP [34], and HSO [35] that were described in our sense similarity experiments (cf. Section 6.2). We also report the results for the Wikipedia-based Explicit Semantic Analysis [27, ESA] and for the concept vector-based measure of Zesch and Gurevych [90], which utilizes the glosses in WordNet (ZG-07) or Wiktionary (ZMG-08) for constructing concept vectors (results for these measures were available on the RG-65 dataset only). As for ESA we used the implementation provided in [146]. In addition, on the RG-65 dataset, we report results for the random walk based techniques of Hughes and Ramage [94] and Agirre et al. [93] that are closest to our approach in respect of their usage of random walks on semantic networks, but which do not involve our alignment-based disambiguation. We also computed the performance of two distributional semantic models, Word2vec [86] and PMI-SVD, for each of which we performed experiments with the best corresponding model obtained by Baroni et al. [81].[15] Word2vec is based on neural network context prediction models [86], whereas PMI-SVD is a traditional cooccurrence based vector wherein weights are calculated by means of Pointwise Mutual Information (PMI) and the vector's dimension is reduced to 500 by singular value decomposition (SVD). In these two systems we used the cosine measure to compute the similarity of a pair of vectors.

6.3.2. Word similarity results

To be consistent with the literature [94,93], we evaluated the performance of our similarity measure according to both Spearman's ρ and Pearson's r correlations. Pearson measures the linear correlation of two variables in terms of the differences in their values, whereas Spearman considers the relative rankings of the values of the two variables. The latter correlation is useful for evaluating systems in a similarity ranking setting where relative scores are important, as opposed to the Pearson correlation, which is more suitable for evaluating a setting in which absolute similarity values matter. We note that, due to the way word similarity datasets are constructed, the Pearson correlation is better able to reflect the reliability of a similarity measure. However, for the sake of completeness, we also report results according to the Spearman correlation.

RG-65 Table 9 shows Spearman's ρ and Pearson's r correlation coefficients with human judgments on the dataset. We present results for ADW when using WordNet (ADW_{WN}) and Wiktionary (ADW_{WKT}). As can be seen, our approach, irrespective of the underlying semantic network or the evaluation approach, achieves competitive results on the dataset. Of the two semantic networks, Wiktionary proves to be the better, achieving considerable Spearman and Pearson correlations of 0.92[16] and 0.91,[17] respectively. This confirms the advantage that our large automatically-constructed Wiktionary graph can provide for measuring word similarity. Among the comparison systems, the WordNet-based measures of LCH [145], WUP [34], and Hughes and Ramage [94] are among the best, indicating the suitability of this resource for the measurement of semantic similarity. The word embeddings approach of Word2vec also attains good performance on both datasets (but lower than ADW_{WKT}). However, we note that the reported performance for Word2vec was obtained upon tuning 48 different configurations of this model on different datasets, including RG-65, by varying different system parameters such as windows size, learning strategy and vector size [81].

An analysis of the results revealed that the better performance of Wiktionary was mainly due to its richer semantic network. For instance, in the RG-65 dataset, the assigned gold similarities for noun pairs *serf-slave* and *furnace-stove* are

[15] http://clic.cimec.unitn.it/composes/semantic-vectors.html.
[16] 99% confidence interval: $0.85 \leq \rho \leq 0.96$.
[17] 99% confidence interval: $0.83 \leq r \leq 0.95$.

Table 10
Spearman's ρ and Pearson's r correlation coefficients with human judgments on the WordSim-353 Similarity (WS-Sim) and YP-130 datasets (99% confidence intervals: $0.61 \leq \rho^{\dagger} \leq 0.82$, and $0.69 \leq r^{\star} \leq 0.86$).

Approach	WS-Sim		YP-130	
	ρ	r	ρ	r
ESA [27]	0.53	0.45	0.26	0.28
WUP [34]	0.54	0.45	0.67	0.77
LCH [145]	0.58	0.56	0.66	0.76
HSO [35]	0.59	0.59	0.62	0.76
PMI-SVD [81]	0.66	0.68	0.30	0.12
Word2vec [86]	**0.78**	**0.76**	0.50	0.32
ADW_{WN}	0.72	0.70	0.71	***0.79**
ADW_{WKT}	0.75	0.72	†**0.73**	0.74

3.46 and 3.11, respectively (in the scale of 0 to 4). The two nouns *serf* and *slave* are connected through different Wiktionary senses such as $bondman_n^1$, $freeman_n^1$, and $helot_n^2$.[18] Similarly, *furnace* and *stove* are connected by multiple word senses, including $fire\_box_n^3$, $damper_n^1$, and $athanor_n^1$.[19] However, the WordNet graph does not provide such a rich set of connections between the two words in each pair. The rich set of connections between these two word pairs in the Wiktionary graph leads to relatively higher calculated similarity scores in comparison to the corresponding mean calculated score μ on the dataset: respectively, $\mu + 0.09$ and $\mu + 0.15$ for Wiktionary compared to $\mu + 0.02$ and $\mu + 0.03$ for WordNet. However, the WordNet graph benefits from having synsets as individual nodes. Therefore, words that are defined as synonyms in the same synset are assigned the maximum score of 1.0 (for instance, *midday-noon* and *gem–jewel* are assigned the respective similarity values of 0.82 and 0.8 when the Wiktionary graph is used whereas both pairs are synonymous in WordNet and their similarities are computed as 1.0 when the WordNet graph is used).

WS-Sim and YP-130 Table 10 lists the results on the WS-Sim (left side) and YP-130 (right side) datasets. Similarly to the RG-65 dataset, ADW proves competitive on these two datasets, achieving the best performance on the verb similarity task on the YP-130 dataset. The best performance on the WS-Sim dataset is obtained by Word2vec ($\rho = 0.78$[20] and $r = 0.76$[21]). However, as also noted earlier, the reported performance for Word2vec was obtained upon tuning its models on different datasets, including WS-Sim. Of the two semantic networks, *WKT* proves the more suitable for noun similarity by attaining better results on the WS-Sim dataset, an outcome that was also observed on the RG-65 dataset. As for verb similarity, the higher Pearson correlation performance of the WordNet graph on the YP-130 shows that this network is more effective at providing accurate similarity judgments between verb pairs. This can be attributed to the manual construction of this resource, as opposed to our automatically constructed WKT network which is prone to noisy edges, particularly for the case of verbs that are characterized by fine-grained sense distinctions and are usually more difficult to disambiguate.

6.3.3. Analysis I: performance on other semantic networks

In Section 4.3 we described our approach for the automatic construction of two semantic networks, namely *W2V* and *DM*, by leveraging two popular distributional semantic models. In addition, we mentioned in Section 4.2 that we also constructed a variant of the Wiktionary graph, called *WKTall*, in which all the content words in the definitions are exploited so as to make a denser semantic network. In order to verify the reliability of our automatically-induced networks for the generation of semantic signatures and also to test whether the richness of ADW's semantic network always results in its better performance, we carried out an evaluation on all these three graphs in the task of word similarity measurement.

Table 11 shows the performance of the variants of ADW using these graphs on our three word similarity datasets. As reference, we also list the results for ADW when using its default networks, i.e., *WN* and *WKT* (the last two rows in the table). As can be seen, ADW attains relatively low performance when semantic signatures are generated on the two automatically-induced semantic networks. The gap is particularly noticeable in the case of the YP-130 dataset, highlighting the weakness of the distributional models in capturing the semantics of verbs effectively. Overall, the results indicate that the manually-crafted relations in *WKT* and *WN* graphs are more suitable for the generation of semantic signatures. For instance, we show in Table 12 ten sampled direct neighbors for three word senses $smartphone_n^1$, $paper_n^1$, and $terminate_v^1$ in the

[18] $bondman_n^1$: "A man who is bound in servitude; a slave or serf.", $freeman_n^1$: "A free man, one who is not a serf or slave.", and $helot_n^2$: "A serf; a slave."
[19] $fire\_box_n^3$: "The internal hearth of a furnace, stove, or heater.", $damper_n^1$: "A valve or movable plate in the flue or other part of a stove, furnace, etc., used to check or regulate the draught of air.", and $athanor_n^1$: "a furnace or stove, designed and used to maintain uniform heat. Primarily used by alchemists."
[20] 99% confidence interval: $0.70 \leq \rho \leq 0.84$.
[21] 99% confidence interval: $0.67 \leq r \leq 0.83$.

Table 11
The performance of variants of ADW when using the automatically-induced semantic networks, i.e., *W2V* and *DM*, and the Wiktionary *WKTall* graph, in which all the content words in the definitions are exploited for maximizing richness, on our three word similarity datasets. We also list, in the last two rows, the results for our two default graphs, i.e., *WN* and *WKT*, from Tables 9 and 10.

Evaluation	Dataset					
	RG-65		YP-130		WS-Sim	
	ρ	r	ρ	r	ρ	r
W2V	0.63	0.65	0.37	0.44	0.69	0.65
DM	0.78	0.80	0.52	0.47	0.59	0.62
WKTall	0.83	0.86	**0.73**	0.74	0.73	0.64
WN	0.86	0.81	0.71	**0.79**	0.72	0.70
WKT	**0.92**	**0.91**	**0.73**	0.74	**0.75**	**0.72**

Table 12
The top 10 direct neighbors of the three word senses *smartphone*^1_n, *paper*^1_n, and *terminate*^1_v in the *WKT* graph.

smartphone^1_n	paper^1_n	terminate^1_v
mobile_phone^1_n	tissue^3_n	close_out^1_v
BlackBerry^1_n	print^2_v	end^1_v
feature_phone^1_n	cellulose^1_n	close^4_v
dumbphone^1_n	sheet^2_n	finish^1_v
smartbook^1_n	letter^2_n	snap_out_of^1_v
iPhone^1_n	write^9_v	stop^4_v
phablet^1_n	fiber_faced^1_a	abort^8_v
debrick^2_v	litmus_paper^1_n	failure^3_n
thumbboard^1_n	wallpaper^1_n	put_an_end_to^1_v
keitai^1_n	manuscript_paper^1_n	abortion_pill^1_n

WKT graph.[22] A comparison between Table 12 and Tables 2 and 3, which list the ten most related terms to the same three words[23] in the automatically-induced networks, reveals the reasons behind the better performance of our manually-crafted semantic networks: (1) the edges in the automatically-induced networks connect synonyms or highly-similar words only, whereas the manually-crafted networks benefit from a wider set of relations of various types (e.g., *print*^2_v to *paper*^1_n, *debrick*^2_v to *smartphone*^1_n, and *abortion_pill*^1_n to *terminate*^1_v); (2) sense-level distinctions provided by the WordNet and Wiktionary graphs result in more accurate representations of meaning in the semantic signatures.

Moreover, ADW performs significantly better when only hyperlinked words in the definitions of word senses are utilized for the construction of the semantic network (i.e., *WKT* graph), rather than all its content words (i.e., *WKTall* graph). This shows that enriching a semantic network would not always be beneficial, as adding edges that carry less specific information (which is more likely to occur for non-hyperlinked words in the definitions) to the network can result in lower-quality semantic signatures, leading to lower performance in the similarity measurement task.

6.3.4. Analysis II: the role of disambiguation

In order to get more insight into the effect of ADW's disambiguation step on the final performance when measuring the similarity of word pairs, we carried out a series of experiments where no disambiguation was performed on the two linguistic items. In the absence of the disambiguation step, a semantic signature for a linguistic item is generated by initializing the PPR algorithm from all the nodes (i.e., synsets) associated with that linguistic item, rather than the specific disambiguated synsets obtained as a result of the alignment-based disambiguation (see Section 5.1). In other words, the personalization vector $\mathcal{S}^0$ in Equation (2) is constructed by uniformly distributing the probability mass among all the corresponding dimensions of all senses of all the content words in the item. Note that in this case the semantic signature of a linguistic item is generated independently of its paired linguistic item.

We show in Table 13 the performance of our system with and without the disambiguation step and, for the sake of comparison, the results that were also presented earlier for the standard setup of our system in Tables 9 and 10. In addition to our two default semantic networks, i.e., *WN* and *WKT*, we also show results for the *WKTall* variant of the Wiktionary graph. As can be seen from the table, the disambiguation phase results in a consistent improvement according to the Pear-

[22] *smartphone*^1_n: "A mobile phone with more advanced features and greater computing capacity than a featurephone.", *paper*^1_n: "A sheet material used for writing on or printing on (or as a non-waterproof container), usually made by draining cellulose fibres from a suspension in water.", and *terminate*^1_v: "To finish or end."

[23] Note that the graph nodes in the automatically-induced networks represent words as opposed to the WordNet and Wiktionary graphs in which individual nodes are word senses or concepts.

Table 13
System performance in terms of Spearman's ρ and Pearson's r correlations with and without the disambiguation step on three word similarity datasets. In addition to our default graphs, i.e., *WN* and *WKT*, we also report results for the *WKTall* variant of the Wiktionary graph.

Disamb.	Dataset											
	RG-65				YP-130				WS-Sim			
	ρ		r		ρ		r		ρ		r	
	yes	no	yes	no	yes	no	yes	no	yes	no	yes	no
ADW_{WN}	**0.86**	0.81	**0.81**	0.71	0.71	**0.76**	**0.79**	0.71	0.72	**0.77**	**0.70**	0.69
ADW_{WKT}	0.92	**0.93**	**0.91**	0.84	0.73	**0.78**	**0.74**	0.70	0.75	**0.77**	**0.72**	0.66
ADW_{WKTall}	**0.83**	0.79	**0.86**	0.81	0.73	**0.78**	**0.74**	0.70	**0.73**	0.72	**0.64**	0.55

Table 14
Sample verb and noun pairs from the YP-130 and RG-65 datasets, respectively. Pairs are sampled according to their gold similarity scores (Gold in the table) from high, medium, and low ranges. We show similarity judgments made by ADW with and without the disambiguation phase and for two graphs, i.e., *WN* and *WKT*.

YP-130						
Gold	Word pair		*WN*		*WKT*	
			yes	no	yes	no
4.0	brag	boast	1.00	0.83	0.98	0.75
3.8	consume	eat	1.00	0.62	0.96	0.54
2.5	twist	interwine	0.73	0.46	0.71	0.47
1.2	swing	crash	0.52	0.43	0.43	0.34
0.3	postpone	show	0.44	0.40	0.35	0.34
0.0	correlate	levy	0.38	0.37	0.34	0.33

RG-65						
Gold	Word pair		*WN*		*WKT*	
			yes	no	yes	no
3.9	car	automobile	1.00	0.66	0.64	0.52
3.9	gem	jewel	1.00	0.77	0.66	0.49
2.7	brother	monk	0.66	0.47	0.51	0.37
2.4	crane	implement	0.46	0.40	0.38	0.36
0.4	glass	magician	0.44	0.38	0.29	0.33
0.0	rooster	voyage	0.40	0.40	0.30	0.31

son correlation on all the three datasets and all the three semantic graphs. This confirms the role of the alignment-based disambiguation phase in accurately estimating the extent of semantic similarity. However, the Spearman correlation performance is not as consistent, as the disambiguation step proves beneficial on the RG-65 dataset, but harmful on the YP-130 and WS-Sim datasets. As also noted earlier in Section 6.3.2, we argue that the Spearman correlation is not as conclusive on the similarity measurement datasets. In fact, the Spearman correlation is more suitable for measuring the performance of a system in ranking a comparable set of pairs according to their similarity. And such ranking was not among the original intended purposes when constructing these datasets in which the annotation has been carried out for individual word pairs separately rather than in relation to each other. In contrast, Pearson correlation measures the performance of systems in respect of their ability to provide judgments that are close to those of humans.

We show in Table 14 sample pairs from the YP-130 and RG-65 datasets, along with the judgments ADW made for them with and without the disambiguation phase. As can be seen, disambiguation enables ADW to obtain more accurate judgments for word pairs, capturing synonymy for pairs such as *consume* and *eat* with high similarity scores. In contrast, when no disambiguation is applied, a word is taken as a conflation of all its senses, resulting in less accurate estimates of similarity.

It is also noteworthy that our system assigns scores in the range 0.3–0.4 even for unrelated word pairs with near-zero gold similarity scores. The reason behind this over-compensation is that two semantic signatures, even if unrelated, have overlapping dimensions. In fact, for the extreme case of two semantic signatures that have their dimensions in complete inverse order, the numerator in equation (8) is equal to one. The denominator is independent of the rankings in the two signatures and can be estimated as:

$$\ln(|H|) + \gamma + \frac{1}{2|H|} - \frac{1}{12|H|^2} + \frac{1}{120|H|^4} - \ldots, \tag{9}$$

where $|H|$ is the intersection of all non-zero dimensions in the two signatures and γ is the Euler–Mascheroni constant which is approximately equal to 0.577. Since we experiment with untruncated vectors, $|H|$ is equal to the size of the semantic network. For a semantic network with the number of nodes in the order of hundreds of thousands (which is approximately the case for our two semantic networks), the denominator in equation (8) is estimated to be approximately 6.5. Therefore, in our setting, the minimum value of WO, which can happen for the case of two inversely ordered signatures, is around 0.15. This explains the over-compensation of similarity scores for unrelated word pairs in Table 14.

6.4. Experiment 3: sentence-level similarity

So far, the experiments we have reported in this article were those carried out to assess the performance of our system when measuring the semantic similarity of sense and word pairs. In this section, we focus on the highest linguistic level, i.e., the text level. Measuring the semantic similarity of sentence pairs, a task usually referred to as Semantic Textual Similarity

Table 15
Statistics of the provided datasets for the SemEval-2012 Semantic Textual Similarity Task. In the last row we provide the inter-annotator agreements (ITA) for each dataset as reported in [152].

Dataset	MSRvid	MSRpar	SMTeuroparl	OnWN	SMTnews
Training	750	750	734	–	–
Test	750	750	459	750	399
ITA	0.87	0.71	0.53	0.63	0.56

(STS), plays an important role in a wide variety of NLP applications such as machine translation, question answering, and summarization. Given a pair of sentences, the goal in this task is to automatically judge their semantic similarity. This judgment should ideally be close to the judgment made by human annotators. As our benchmark for this level we take the datasets from the STS task in SemEval-2012 [107, STS-12]. The STS-12 task defined a similarity scale ranging from 0 to 5, where 5 denotes that the pair of sentences are identical in meaning. The performance of a system in this task is measured by calculating the linear correlation between the output judgments and human-assigned similarity scores on the entire set of sentence pairs.

Most of the participating systems in the task are supervised systems that utilize the provided training data in order to learn regression models, with the goal of producing scores that are closest to those assigned by human annotators. The systems are generally a combination of the similarity judgments made by several lexico-semantic similarity measures. However, in such a setting, the role of individual features in the overall system performance is not explicitly distinguishable, since the regression model and the complementary nature of different similarity measures can play an important role in the final similarity judgment. In addition, in a supervised setting, the performance of the system, when measured in terms of linear correlation, depends strongly on the domain and characteristics of the training data [29,149–151]. Therefore, we carry out our experiments in an unsupervised setting and on a single-measure basis so as to be able to draw a direct conclusion on the performance of each similarity measure independently of the regression model or other implicit factors. To this end, we compare our approach against state-of-the-art similarity measures that are used independently for judging similarities of sentence pairs.

6.4.1. Comparison systems

For our experiments we picked as benchmark five of the best-performing similarity measures utilized in the UKP system, the top-ranking system in STS-12. These measures are implemented in the DKProSimilarity package [29,146]. As a representative of measures that are based on vector space models, we selected Explicit Semantic Analysis [27, ESA]. ESA represents a term as a high-dimensional vector wherein the dimensions are articles in Wikipedia and the weights are the *tf–idf* values of that term in the corresponding article. The approach was later extended to other lexical-semantic resources such as WordNet and Wiktionary [90]. In the experiments, we obtain results for ESA when the three above-mentioned resources were used for the construction of the vector space [29,28].

We also compared ADW against five lexical resource-based measures, WUP [34], JCN [38], LIN [39], LCH [36], and RES [37], which were also used in our sense- and word-level experiments. In order to compute the similarity of text pairs using these five concept similarity measures, we used the aggregation method proposed by Corley and Mihalcea [26]. This approach computes the similarity between a pair of sentences T_1 and T_2 by finding, for each word w in T_1, the most similar word in T_2:

$$\text{sim}(T_1, T_2) = \frac{\sum_{w \in T_1} \text{maxSim}(w, T_2)\, \text{idf}(w)}{\sum_{w \in T_1} \text{idf}(w)}, \tag{10}$$

where $\text{maxSim}(w, T_2)$ returns the similarity value between w and its most similar word in T_2 and $\text{idf}(w)$ is the inverse document frequency of the term w. The final similarity score is computed as the average similarity in the two directions, i.e., the average of $\text{sim}(T_1, T_2)$ and $\text{sim}(T_2, T_1)$.

6.4.2. Datasets

The STS-12 task provided five test sets, namely MSRpar, SMTeuroparl, SMTnews, MSRvid, and OnWN. The first three datasets belong to the newswire genre, whereas the other two are generic. For each of these datasets, we show in Table 15 the number of sentence pairs in the training and test datasets. OnWN and SMTnews were not provided with any accompanying training data [107]. The gold standard similarity judgments for each sentence pair is an average of scores assigned by five human annotators. The annotation task was carried out through crowdsourcing, while certain post-hoc validations were deployed to ensure quality. In the last row in Table 15, we also report the inter-annotator agreements (ITA) for each dataset: the scores range from 0.53 (SMTeuroparl) to 0.87 (MSRvid) denoting the varying difficulties of similarity judgments for each dataset.

Table 16
Variants of our system in the sentence-level experiment.

System variant	Semantic network	OOV handling
ADW_{WN}	WordNet	None
$ADW_{WN:I}$	WordNet	Direct OOV injection
$ADW_{WN:M}$	WordNet	Wikipedia-assisted mapping
ADW_{WT}	Wiktionary	None
$ADW_{WT:I}$	Wiktionary	Direct OOV injection

Table 17
Performance of variants of ADW together with other similarity measures on the five datasets of the SemEval-2012 Semantic Textual Similarity task in terms of the Pearson (r) and Spearman (ρ) correlations. The right-most columns show the average (Avg_{macro}) and the weighted average (Avg_{micro}) performance across the five datasets. We show the performance for the five settings of ADW (see Table 16). Results are provided for Explicit Semantic Analysis (ESA) based on Wikipedia (*WP*), Wiktionary (*WT*), and WordNet (*WN*).

System	MSRvid		MSRpar		SMTeuroparl		OnWN		SMTnews		Avg_{micro}		Avg_{macro}	
	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r
WUP	0.65	0.64	0.39	0.44	0.22	0.21	0.52	0.55	0.25	0.28	0.44	0.45	0.40	0.42
JCN	0.65	0.65	0.39	0.44	0.20	0.20	0.50	0.53	0.24	0.26	0.43	0.45	0.40	0.41
LCH	0.68	0.67	0.39	0.45	0.19	0.18	0.50	0.53	0.24	0.27	0.42	0.45	0.40	0.43
LIN	0.70	0.70	0.41	0.45	0.23	0.23	0.54	0.57	0.25	0.28	0.46	0.48	0.42	0.45
RES	0.74	0.73	0.41	0.46	0.25	0.25	0.55	0.59	0.27	0.30	0.48	0.50	0.45	0.47
ESA_{WN}	0.78	0.78	0.28	0.25	0.30	0.22	0.62	0.59	0.22	0.13	0.47	0.43	0.44	0.39
ESA_{WT}	0.77	0.77	0.36	0.34	0.30	0.22	**0.66**	**0.65**	0.33	0.27	0.51	0.48	0.48	0.45
ESA_{WP}	0.75	0.74	0.43	0.44	0.38	0.48	0.62	0.62	0.33	0.40	0.53	0.56	0.50	0.54
ADW_{WN}	**0.80**	**0.80**	0.42	0.45	**0.54**	0.54	0.64	0.63	0.33	0.38	0.58	0.60	0.55	0.56
$ADW_{WN:I}$	**0.80**	**0.80**	**0.51**	**0.56**	**0.54**	**0.55**	0.61	0.63	0.34	0.40	**0.61**	**0.63**	**0.56**	**0.59**
$ADW_{WN:M}$	**0.80**	**0.80**	0.43	0.46	0.53	0.54	0.64	0.63	0.33	0.38	0.59	0.60	0.55	0.56
ADW_{WT}	**0.80**	**0.80**	0.34	0.35	0.51	0.49	0.60	0.54	**0.43**	0.44	0.56	0.55	0.54	0.52
$ADW_{WT:I}$	**0.80**	**0.80**	0.48	0.51	0.52	0.50	0.59	0.54	0.42	**0.45**	0.60	0.60	**0.56**	0.56

6.4.3. Experimental setup

As explained earlier, we performed sentence-level experiments in an unsupervised setting, and hence there was no need to train regression models for similarity judgments. For this reason, we combined the training and test datasets (when a training set was available, see Table 15) and performed an evaluation on the whole dataset.

6.4.4. System variants

We proposed two different approaches for handling out-of-vocabulary words in Section 5.4, i.e., direct OOV injection and Wikipedia-assisted OOV mapping. Regarding our sentence-level experiments, we provide results when utilizing these approaches. We show in Table 16 different variants of our system in the sentence-level experiment. Note that the Wikipedia-assisted OOV mapping approach cannot be applied when semantic signatures are generated on the Wiktionary semantic network (cf. Section 5.4.2).

6.4.5. Evaluation

We followed the STS-12 task and evaluated the performance of sentence similarity measures in terms of correlation with the gold standard scores. However, in addition to the Pearson correlation, which was used in STS-12, we also provide results in terms of the Spearman rank correlation.

6.4.6. STS results

We show in Table 17 the performance of our measure, ADW, together with the six other similarity measures on the five datasets of STS-12. The rightmost columns in the table show the average performance on the five datasets (Avg_{macro}) and the average performance weighted by the number of pairs in each dataset (Avg_{micro}). ESA_{WP}, ESA_{WT}, ESA_{WN} correspond to ESA when, respectively, Wikipedia, Wiktionary, and WordNet were used for building up the vector space. As can be seen from the table, our measure provides competitive results on all the datasets, while achieving the best overall performance when semantic signatures were generated using the WordNet graph and injected directly with OOV words (i.e., $ADW_{WN:I}$) according to both Spearman (Avg_{micro} 0.61[24] and Avg_{macro} 0.56[25]) and Pearson (Avg_{micro} 0.63[26] and Avg_{macro} 0.59[27]) correlations.

[24] 99% confidence interval: $0.59 \leq \rho \leq 0.63$.
[25] 99% confidence interval: $0.54 \leq \rho \leq 0.59$.
[26] 99% confidence interval: $0.61 \leq r \leq 0.65$.
[27] 99% confidence interval: $0.56 \leq r \leq 0.61$.

Table 18
Some of the least accurate judgments made by ADW on the sentence pairs of the MSRpar and SMTnews datasets. Content words that are defined in WordNet are shown in bold and the OOV words are highlighted in italics. The gold scores are scaled from the original scale of [0, 5] to [0, 1].

	Gold	$ADW_{WN:I}$	Sentence pair
(a)	0.20	0.79	*side-1:* To **reach** *John A. Dvorak*, who **covers Kansas**, **call** (816) 234-7743 or **send e-mail** to jdvorak@kctar.com. *side-2:* To **reach** *Brad Cooper, Johnson County* **municipal reporter**, **call** (816) 234-7724 or **send e-mail** to bcooper@kcstar.com.
(b)	0.90	0.59	*side-1:* This **tendency extends deeper** than *headscarves*. *side-2:* This **trend goes well beyond simple scarves**.
(c)	0.60	1.00	*side-1:* He **did**, but the **initiative did** not **get** very **far**. *side-2:* What he has **done** without the **initiative goes** too **far**.
(d)	0.76	1.00	*side-1:* **Last year** he was **wanted** for **murder**. *side-2:* **Last year** it was **required** for **murder**.
(e)	0.32	0.74	*side-1:* **Oracle shares fell** 27 **cents**, or 2 **percent**, to $13.09. *side-2:* **Oracle shares** also **rose** on the **news**, up 15 **cents** to $13.51 on the **Nasdaq**.
(f)	0.40	0.92	*side-1:* The 30-year **bond** US30YT=RR **jumped** 20/32, **taking** its **yield** to 4.14 **percent** from 4.18 **percent**. *side-2:* The 30-year **bond** US30YT=RR **dipped** 14/32 for a **yield** of 4.26 **percent** from 4.23 **percent**.

The OOV handling proves to be beneficial on most datasets and for both semantic networks. On the WordNet graph, among the two OOV handling approaches, direct OOV injection attains higher performance. Specifically, the approach leads to the statistically significant[28] Pearson correlation improvements of 0.03 and 0.05 (Avg_{micro}) over the system with no OOV handling on the WordNet and Wiktionary networks, respectively. Among the five datasets, the largest gain by OOV injection is obtained on the MSRpar dataset, indicating the high number of WordNet OOV words in the dataset. Specifically, 14% of the nouns in the dataset are not defined in WordNet's vocabulary, words that are mostly proper nouns in this newswire domain dataset. The second dataset with the highest OOV noun proportion is the SMTnews dataset with 8%, followed by OnWN with 5%. SMTeuroparl (3%) and MSRvid (1%) are among the easiest datasets with respect to OOV proportion.

ESA provides generally good results on the five datasets, with ESA_{WP} (i.e., ESA based on Wikipedia) proving to be the best among the three versions. ESA_{WT} obtains the best performance on OnWN, a dataset containing sense glosses from OntoNotes 4.0 [153] and WordNet 3.1 [31]. The ESA_{WT} system is trained on a similar dataset, i.e., word sense definitions extracted from Wiktionary. This makes the system particularly effective on the OnWN dataset. However, ESA_{WT} does not provide a reliable performance across other datasets, especially in the newswire domain, i.e., SMTeuroparl, SMTnews, and MSRpar. ESA_{WN} performs best among different versions of ESA on the MSRvid dataset, while lagging behind on the other four datasets. This performance variation can be attributed to the different nature of the MSRvid dataset, which comprises short sentences that do not contain many WordNet OOV words or proper nouns. Lexical resource-based measures, i.e., WUP, JCN, LIN, LCH, and RES, despite proving reliable at word similarity, generally fall behind ESA_{WP} on all datasets. Among them, RES proves to be the most reliable approach for sentence similarity measurement, providing the second best overall performance.

Discussion I: analysis of lower performance on MSRpar and SMTnews Apart from on the two datasets with the highest OOV proportions, i.e., MSRpar and SMTnews, ADW's performance comes close to the reported inter-annotator agreement (see ITA in Table 15), indicating the reliability of ADW for accurate similarity measurement. We attribute the relatively low performance on MSRpar and SMTnews mainly to the high number of OOV words in these two datasets. To provide a better analysis of this attribution, and demonstrate some of the other reasons behind this lower performance, we show in Table 18 six sentence pairs from the two datasets for which $ADW_{WN:I}$'s judgments are highly divergent from the gold judgments. In examples (a) and (b), the neglect of the OOV words (highlighted in italics) is mainly responsible for the less accurate similarity estimations made by ADW. Note that almost all the OOV words in the two sentence pairs are not defined in Wiktionary and also do not match across the two sentences. Hence, our OOV handling techniques would also fall short in assisting ADW to provide a more accurate similarity computation for these pairs. The sentence pairs (c) and (d) represent cases in which non-content words and the syntactic structure of the sentences have significantly affected their semantic similarity. The syntax of a sentence is an important factor, which is often not taken into account during similarity measurement, as is also the case for ADW. The last two examples (e) and (f) demonstrate an important characteristic of the MSRpar dataset: a significant presence of numbers. Apart from the OOV words and the grammatical structure of these two sentence pairs, numerical items played an important role in the semantics of the sentences and accordingly in their similarity scores. Lack of means for modeling numerical items and special characters (e.g., see the sentence pair (a) for phone numbers and email addresses) is one more shortcoming of ADW and most other similarity measures.

[28] According to z-test at $p < 0.05$.

Discussion II: disambiguating longer textual items As we mentioned in Section 5.3 the disambiguation step is particularly suitable for short textual items, such as words or phrases, as an implicit disambiguation takes place when modeling longer textual items such as sentences. We argued that the disambiguation step might not be as effective for long textual items. In order to verify this, we carried out our sentence-level experiments with a variant of our system that did not involve the disambiguation step. As we also described in our analysis in the word level, in the absence of the disambiguation step, a semantic signature is constructed by initializing the PPR algorithm from all nodes associated with all the senses of the content words in the linguistic item. Experiments were carried out on the five sentence-level datasets and for the three variants of our WordNet-based system, i.e., ADW_{WN}, $ADW_{WN:1}$, and $ADW_{WN:M}$ and the two variants of the Wiktionary-based system, i.e., ADW_{WT}, $ADW_{WT:1}$. We observed that in all the variants and according to both evaluation measures, the differences in the performance values with and without the disambiguation step were not statistically significant. This confirms our hypothesis that, when there is enough context in a linguistic item, which is the case in sentence similarity, an implicit disambiguation takes place during the generation of semantic signatures. However, note that when the disambiguation step is involved, our system provides the advantage of having, as a byproduct of similarity measurement, the two linguistic items explicitly disambiguated with respect to each other according to the adopted semantic network.

6.5. Summary of the results

We performed a series of experiments in order to evaluate our similarity measure at three different linguistic levels, i.e., senses, words, and sentences. Here, we summarize the results and findings of our experiments:

- ADW proved to be highly flexible and reliable by providing competitive results on several evaluation benchmarks in three linguistic levels, outperforming state-of-the-art approaches (that usually focus on a specific level) on most gold-standard datasets.
- Our method for comparing semantic signatures, Weighted Overlap, provides an effective technique for comparing semantic signatures, consistently outperforming cosine similarity, Jensen–Shannon divergence, and Rank-Biased Overlap on different datasets.
- We demonstrated that the alignment-based disambiguation phase of our approach provides a consistent system improvement in terms of Pearson correlation when measuring the semantic similarity of short textual items on all the three experimented word-level datasets.
- We found that the automatically-constructed Wiktionary graph can act as a reliable replacement for the manually-crafted WordNet graph when measuring semantic similarity of textual items, proving both the flexibility of our similarity measure and the potential of collaborative resources [112] for the construction of semantic networks and semantic similarity measurement.
- Direct OOV injection proved to be an effective technique for handling out-of-vocabulary words by providing statistically significant improvements for both WordNet and Wiktionary graphs in our sentence-level experiment. Additionally, the simple OOV injection procedure usually leads to better performance in comparison to the Wikipedia-assisted OOV mapping. However, we note that as far as OOV handling is concerned there is still room for improvement, something that we plan to investigate in future work.

7. Conclusions

In this article we have described a unified graph-based approach for measuring the semantic similarity of arbitrary pairs of linguistic items. Our approach leverages random walks on semantic networks in order to obtain rich unified representations for different kinds of input linguistic items: senses, words, and texts. Three sets of experiments were carried out, at three different linguistic levels, in order to evaluate the proposed similarity measure on multiple datasets and settings. We demonstrated that the same unified approach can outperform state-of-the-art techniques, which are often limited in their type of input, across different linguistic levels, providing a reliable basis for measuring the similarity of arbitrary linguistic items for downstream NLP and AI applications. We have shown how collaboratively-constructed resources such as Wiktionary can be transformed into semantic networks for reliable similarity measurement. Moreover, we proposed two new techniques for handling words that are not defined in the lexical resource: one is based on a simple backing off to string similarity and the other exploits a large-scale taxonomy. Improvements were obtained when using both techniques, although the former of the two, the simpler approach, proved to be the more effective. We remark that our similarity measurement approach, in addition to being reliable, unified, and flexible, has an advantage over conventional similarity measurement techniques in that it provides the disambiguated linguistic items as a byproduct of similarity measurement.

As future work, we plan to try our similarity measure on semantic networks obtained from other collaboratively-constructed resources, such as Wikipedia and OmegaWiki, or from thesauri such as Roget's [44]. The Wikipedia graph is expected to be particularly suitable for measuring the similarity between texts as it provides a remarkable coverage of proper nouns and domain-specific terms. We also intend to extend the approach so as to measure the semantic similarity of linguistic items across languages in the lines of [53], a goal that can be achieved by the help of large multilingual semantic networks such as BabelNet [18].

Finally, we are releasing the Java implementation of our system at https://github.com/pilehvar/adw/, providing the research community with an easy-to-use tool for performing semantic similarity of arbitrary linguistic items out of the box. We also provide an online demo of ADW and the pre-computed semantic signatures for all the synsets in WordNet 3.0 at http://lcl.uniroma1.it/adw/. We hope this framework will provide a reliable basis for similarity measurement in different NLP applications and foster further research in the development of unified similarity techniques.

Acknowledgements

The authors gratefully acknowledge the support of the ERC Starting Grant MultiJEDI No. 259234. We also thank the three anonymous reviewers who provided helpful suggestions to improve the paper.

Appendix A. Proposition 1

Consider the vector sequence given by the following recursive rule:

$$v_t = c_1\, v_0 + c_2\, \mathbf{A}\, v_{t-1} \tag{A.1}$$

where $c_1, c_2 \in \mathbb{R}$ and $\mathbf{A} = (a_{i,j})$ is an $l \times l$ matrix. Let $v_0^1, \ldots, v_0^m$ and $w_0 = \frac{\sum_{i=1}^m v_0^i}{m}$ be the respective initial vectors of the sequences $\{v_n^1\}, \ldots, \{v_n^m\}$ and $\{w_n\}$, defined by the above recursive rule. Then,

$$w_t = \frac{\sum_{i=1}^m v_t^i}{m} \quad \forall t \in \mathbb{N} \tag{A.2}$$

Proof. We prove the proposition by induction on t:

- *Base case:* $w_0 = \frac{\sum_{i=1}^m v_0^i}{m}$ by the definition of w_0.
- Assuming the statement holds true for $t-1$, i.e., $w_{t-1} = \frac{\sum_{i=1}^m v_{t-1}^i}{m}$ (induction hypothesis), we will prove it for t.
  Based on the recursive rule of the sequence (equation (A.1)), the kth element of the vector v_t^i, i.e., $v_t^i(k)$, can be obtained as follows:

$$v_t^i(k) = c_1 v_0^i(k) + c_2 \sum_{j=1}^{l} a_{kj} v_{t-1}^i(j) \qquad \forall k = 1, \ldots, l \tag{A.3}$$

  Hence,

$$\sum_{i=1}^{m} v_t^i(k) = \sum_{i=1}^{m} \left(c_1 v_0^i(k) + c_2 \sum_{j=1}^{l} a_{kj} v_{t-1}^i(j) \right) \tag{A.4}$$

  Likewise,

$$w_t(k) = c_1 w_0(k) + c_2 \sum_{j=1}^{l} a_{kj} w_{t-1}(j) \qquad \forall k = 1, \ldots, l \tag{A.5}$$

  By applying the commutative and associative properties of summation, the definition of w_0 and the induction hypothesis:

$$\begin{aligned} w_t(k) &= c_1 w_0(k) + c_2 \sum_{j=1}^{l} a_{kj} w_{t-1}(j) = c_1 \sum_{i=1}^{m} \frac{v_0^i(k)}{m} + c_2 \sum_{j=1}^{l} a_{kj} \frac{\sum_{i=1}^m v_{t-1}^i(j)}{m} \\ &= \sum_{i=1}^{m} \frac{c_1 v_0^i(k)}{m} + \sum_{i=1}^{m} \frac{c_2 \sum_{j=1}^{l} a_{kj} v_{t-1}^i(j)}{m} = \frac{\sum_{i=1}^{m} \left(c_1 v_0^i(k) + c_2 \sum_{j=1}^{l} a_{kj} v_{t-1}^i(j) \right)}{m} \quad \forall k = 1, \ldots, l \end{aligned} \tag{A.6}$$

  From equation (A.4), we have:

$$w_t(k) = \frac{\sum_{i=1}^{m}\left(c_1 v_0^i(k) + c_2 \sum_{j=1}^{l} a_{kj} v_{t-1}^i(j)\right)}{m} = \frac{\sum_{i=1}^{m} v_t^i(k)}{m} \quad \forall k = 1, \ldots, l \tag{A.7}$$

Therefore $w_t = \frac{\sum_{i=1}^{m} v_t^i}{m}$; hence, the induction holds and we have proven the proposition. □

References

[1] A. Lavie, M.J. Denkowski, The Meteor metric for automatic evaluation of Machine Translation, Mach. Transl. 23 (2–3) (2009) 105–115.
[2] O. Glickman, I. Dagan, Acquiring lexical paraphrases from a single corpus, in: Proceedings of Recent Advances in Natural Language Processing, Borovets, Bulgaria, 2003, pp. 81–90.
[3] I. Androutsopoulos, P. Malakasiotis, A survey of paraphrasing and textual entailment methods, J. Artif. Intell. Res. 38 (2010) 135–187.
[4] M. Marelli, L. Bentivogli, M. Baroni, R. Bernardi, S. Menini, R. Zamparelli, SemEval-2014 task 1: evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment, in: Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval 2014, Dublin, Ireland, 2014, pp. 1–8.
[5] A. Hliaoutakis, G. Varelas, E. Voutsakis, E.G. Petrakis, E. Milios, Information retrieval by semantic similarity, Int. J. Semantic Web Inf. Syst. 2 (3) (2006) 55–73.
[6] A. Otegi, X. Arregi, O. Ansa, E. Agirre, Using knowledge-based relatedness for information retrieval, Knowl. Inf. Syst. (2014) 1–30.
[7] M. Mohler, R. Bunescu, R. Mihalcea, Learning to grade short answer questions using semantic similarity measures and dependency graph alignments, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, HLT'11, Portland, Oregon, vol. 1, 2011, pp. 752–762.
[8] A. Fader, L. Zettlemoyer, O. Etzioni, Open question answering over curated and extracted knowledge bases, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 1156–1165.
[9] D. Wang, T. Li, S. Zhu, C. Ding, Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization, in: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'08, Singapore, Singapore, 2008, pp. 307–314.
[10] D. McCarthy, R. Navigli, The English lexical substitution task, Lang. Resour. Eval. 43 (2) (2009) 139–159.
[11] O. Biran, S. Brody, N. Elhadad, Putting it simply: a context-aware approach to lexical simplification, in: Proceedings of 49th Annual Meeting of the Association for Computational Linguistics, Portland, Oregon, 2011, pp. 496–501.
[12] E.M. Voorhees, Query expansion using lexical-semantic relations, in: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'94, 1994, pp. 61–69.
[13] R. Snow, S. Prakash, D. Jurafsky, A.Y. Ng, 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, 2007, pp. 1005–1014.
[14] R. Navigli, 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, COLING–ACL, Sydney, Australia, 2006, pp. 105–112.
[15] R. Navigli, Word Sense Disambiguation: a survey, ACM Comput. Surv. 41 (2) (2009) 1–69.
[16] R. Navigli, P. Velardi, A. Cucchiarelli, F. Neri, Extending and enriching WordNet with OntoLearn, in: Proceedings of the 2nd Global WordNet Conference 2004, GWC 2004, Brno, Czech Republic, 2004, pp. 279–284.
[17] M. Ruiz-Casado, E. Alfonseca, P. Castells, Automatic assignment of Wikipedia encyclopedic entries to WordNet synsets, in: Advances in Web Intelligence, in: Lecture Notes in Computer Science, vol. 3528, Springer Verlag, Lodz, Poland, 2005.
[18] R. Navigli, S.P. Ponzetto, BabelNet: the automatic construction, evaluation and application of a wide-coverage multilingual semantic network, Artif. Intell. 193 (2012) 217–250.
[19] M. Matuschek, I. Gurevych, Dijkstra-WSA: a graph-based approach to word sense alignment, Trans. Assoc. Comput. Linguist. 1 (2013) 151–164.
[20] M.T. Pilehvar, R. Navigli, A robust approach to aligning heterogeneous lexical resources, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, USA, 2014, pp. 468–478.
[21] G. Salton, A. Wong, C.S. Yang, A vector space model for automatic indexing, Commun. ACM 18 (11) (1975) 613–620.
[22] A.G. Maguitman, F. Menczer, H. Roinestad, A. Vespignani, Algorithmic detection of semantic similarity, in: Proceedings of 22nd International Conference on World Wide Web, Chiba, Japan, 2005, pp. 107–116.
[23] T. Elsayed, J. Lin, D.W. Oard, Pairwise document similarity in large collections with MapReduce, in: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers, Columbus, Ohio, 2008, pp. 265–268.
[24] P.D. Turney, P. Pantel, From frequency to meaning: vector space models of semantics, J. Artif. Intell. Res. 37 (2010) 141–188.
[25] A. Budanitsky, G. Hirst, Evaluating WordNet-based measures of lexical semantic relatedness, Comput. Linguist. 32 (1) (2006) 13–47.
[26] C. Corley, R. Mihalcea, Measuring the semantic similarity of texts, in: Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment, Ann Arbor, Michigan, 2005, pp. 13–18.
[27] E. Gabrilovich, S. Markovitch, Computing semantic relatedness using Wikipedia-based explicit semantic analysis, in: Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, 2007, pp. 1606–1611.
[28] D. Bär, T. Zesch, I. Gurevych, 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, 2013, pp. 121–126.
[29] D. Bär, C. Biemann, I. Gurevych, T. Zesch, UKP: computing semantic textual similarity by combining multiple content similarity measures, in: Proceedings of SemEval-2012, Montréal, Canada, 2012, pp. 435–440.
[30] M.T. Pilehvar, D. Jurgens, R. Navigli, Align, disambiguate and walk: a unified approach for measuring semantic similarity, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, 2013, pp. 1341–1351.
[31] C. Fellbaum (Ed.), WordNet: An Electronic Database, MIT Press, Cambridge, MA, 1998.
[32] R. Rada, E. Bicknell, Ranking documents with a thesaurus, J. Am. Soc. Inf. Sci. 40 (5) (1989) 304–310.
[33] R. Rada, H. Mili, E. Bicknell, M. Blettner, Development and application of a metric on semantic nets, IEEE Trans. Syst. Man Cybern. 19 (1) (1989) 17–30.
[34] Z. Wu, M. Palmer, Verbs semantics and lexical selection, in: Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics, ACL'94, Las Cruces, New Mexico, 1994, pp. 133–138.
[35] G. Hirst, D. St-Onge, Lexical chains as representations of context for the detection and correction of malapropisms, in: C. Fellbaum (Ed.), WordNet: An Electronic Lexical Database, MIT Press, 1998, pp. 305–332.
[36] C. Leacock, M. Chodorow, Combining local context and WordNet similarity for word sense identification, in: C. Fellbaum (Ed.), WordNet: An Electronic Lexical Database, MIT Press, 1998, pp. 265–283.

[37] P. Resnik, Using information content to evaluate semantic similarity in a taxonomy, in: Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montreal, Canada, vol. 1, 1995, pp. 448–453.
[38] J.J. Jiang, D.W. Conrath, Semantic similarity based on corpus statistics and lexical taxonomy, in: Proceedings of International Conference Research on Computational Linguistics, ROCLING X, Taiwan, 1997, pp. 19–30.
[39] D. Lin, An information-theoretic definition of similarity, in: Proceedings of the Fifteenth International Conference on Machine Learning, San Francisco, CA, 1998, pp. 296–304.
[40] M. Lesk, Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone, in: Proceedings of the 5th Annual Conference on Systems Documentation, Toronto, Ontario, Canada, 1986, pp. 24–26.
[41] S. Banerjee, T. Pedersen, Extended gloss overlap as a measure of semantic relatedness, in: Proceedings of the 18th International Joint Conference on Artificial Intelligence, Acapulco, Mexico, 2003, pp. 805–810.
[42] E. Agirre, O. Lopez, Clustering WordNet word senses, in: Proceedings of Recent Advances in Natural Language Processing, Borovets, Bulgaria, 2003, pp. 121–130.
[43] M.T. Pilehvar, R. Navigli, A large-scale pseudoword-based evaluation framework for state-of-the-art word sense disambiguation, Comput. Linguist. 40 (4) (2014) 837–881.
[44] P.M. Roget, Roget's International Thesaurus, 1st edition, Cromwell, New York, USA, 1911.
[45] J.R.L. Bernard (Ed.), Macquarie Thesaurus, Macquarie, Sydney, Australia, 1986.
[46] H. Kozima, T. Furugori, Similarity between words computed by spreading activation on an English dictionary, in: Proceedings of the Sixth Conference on European Chapter of the Association for Computational Linguistics, EACL'93, Utrecht, The Netherlands, 1993, pp. 232–239.
[47] H. Kozima, A. Ito, Context-sensitive word distance by adaptive scaling of a semantic space, in: Recent Advances in Natural Language Processing: Selected Papers from RANLP 1995, in: Amsterdam Studies in the Theory and History of Linguistic Science: Current Issues in Linguistic Theory, vol. 136, John Benjamins Publishing Company, Tzigov Chark, Bulgaria, 1997, pp. 111–124, Chapter 2.
[48] J. Morris, G. Hirst, Lexical cohesion computed by thesaural relations as an indicator of the structure of text, Comput. Linguist. 17 (1) (1991) 21–48.
[49] M. Jarmasz, S. Szpakowicz, Roget's thesaurus and semantic similarity, in: Proceedings of Recent Advances in Natural Language Processing, Borovets, Bulgaria, 2003, pp. 212–219.
[50] R. Navigli, S.P. Ponzetto, BabelRelate! a joint multilingual approach to computing semantic relatedness, in: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, AAAI-12, Toronto, Canada, 2012, pp. 108–114.
[51] N. Erbs, I. Gurevych, T. Zesch, Sense and similarity: a study of sense-level similarity measures, in: Proceedings of the Third Joint Conference on Lexical and Computational Semantics, *SEM 2014, Dublin, Ireland, 2014, pp. 30–39.
[52] J. Camacho-Collados, M.T. Pilehvar, R. Navigli, NASARI: a novel approach to a semantically-aware representation of items, in: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Denver, Colorado, 2015, pp. 567–577.
[53] J. Camacho-Collados, M.T. Pilehvar, R. Navigli, A unified multilingual semantic representation of concepts, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, Beijing, China, 2015.
[54] L. Lee, Measures of distributional similarity, in: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics, College Park, Maryland, 1999, pp. 25–31.
[55] S. Mohammad, G. Hirst, Distributional measures of semantic distance: a survey, CoRR, arXiv:1203.1858.
[56] J. Reisinger, J.R. Mooney, Multi-prototype vector-space models of word meaning, in: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Los Angeles, California, 2010, pp. 109–117.
[57] S. Reddy, I. Klapaftis, D. McCarthy, S. Manandhar, Dynamic and static prototype vectors for semantic composition, in: Proceedings of 5th International Joint Conference on Natural Language Processing, Chiang Mai, Thailand, 2011, pp. 705–713.
[58] E.H. Huang, R. Socher, C.D. Manning, A.Y. Ng, Improving word representations via global context and multiple word prototypes, in: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers, Jeju Island, Korea, vol. 1, 2012, pp. 873–882.
[59] X. Chen, Z. Liu, M. Sun, A unified model for word sense representation and disambiguation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, Doha, Qatar, 2014, pp. 1025–1035.
[60] I. Iacobacci, M.T. Pilehvar, R. Navigli, SensEmbed: learning sense embeddings for word and relational similarity, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, Beijing, China, 2015.
[61] H. Rubenstein, J.B. Goodenough, Contextual correlates of synonymy, Commun. ACM 8 (10) (1965) 627–633.
[62] D. Yang, D.M.W. Powers, Verb similarity on the taxonomy of WordNet, in: The 3rd International WordNet Conference, GWC-06, Jeju Island, Korea, 2006, pp. 121–128.
[63] L. Finkelstein, G. Evgenly, M. Yossi, R. Ehud, S. Zach, W. Gadi, R. Eytan, Placing search in context: the concept revisited, ACM Trans. Inf. Syst. 20 (1) (2002) 116–131.
[64] Z. Harris, Distributional structure, Word 10 (1954) 146–162.
[65] J. Weeds, D. Weir, Co-occurrence retrieval: a flexible framework for lexical distributional similarity, Comput. Linguist. 31 (4) (2005) 439–475.
[66] T. Landauer, S. Dooley, Latent semantic analysis: theory, method and application, in: Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community, CSCL'02, Boulder, Colorado, 2002, pp. 742–743.
[67] S.C. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas, R.A. Harshman, Indexing by latent semantic analysis, J. Am. Soc. Inf. Sci. 41 (6) (1990) 391–407.
[68] D. Lin, Automatic retrieval and clustering of similar words, in: Proceedings of the 17th Conference on Computational Linguistics, Montreal, Quebec, Canada, 1998, pp. 768–774.
[69] S. Pado, M. Lapata, Dependency-based construction of semantic space models, Comput. Linguist. 33 (2) (2007) 161–199.
[70] K. Erk, S. Padó, A structured vector space model for word meaning in context, in: Proceedings of the 2008 Conference on Empirical Methods in Natural Language, EMNLP, Edinburgh, UK, 2008, pp. 897–906.
[71] K.S. Jones, A statistical interpretation of term specificity and its application in retrieval, J. Doc. 28 (1972) 11–21.
[72] K.W. Church, P. Hanks, Word association norms, mutual information, and lexicography, Comput. Linguist. 16 (1) (1990) 22–29.
[73] S. Evert, The statistics of word cooccurrences: word pairs and collocations, Ph.D. thesis, Universität Stuttgart, 2005, http://elib.uni-stuttgart.de/opus/volltexte/2005/2371.
[74] T.K. Landauer, S.T. Dumais, A solution to Plato's problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge, Psychol. Rev. 104 (2) (1997) 211.
[75] J. Bullinaria, J. Levy, Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and SVD, Behav. Res. Methods 44 (3) (2012) 890–907.
[76] T. Hofmann, Unsupervised learning by probabilistic latent semantic analysis, Mach. Learn. 42 (1) (2001) 177–196.
[77] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[78] M. Steyvers, T. Griffiths, Probabilistic topic models, Handb. Latent Semant. Anal. 427 (7) (2007) 424–440.
[79] M. Yazdani, A. Popescu-Belis, Computing text semantic relatedness using the contents and links of a hypertext encyclopedia: extended abstract, in: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, Beijing, China, 2013, pp. 3185–3189.

[80] J. Turian, L. Ratinov, Y. Bengio, Word representations: a simple and general method for semi-supervised learning, in: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, 2010, pp. 384–394.
[81] M. Baroni, G. Dinu, G. Kruszewski, Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Baltimore, Maryland, 2014, pp. 238–247.
[82] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation, in: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, 1986, pp. 318–362.
[83] J.L. Elman, Finding structure in time, Cogn. Sci. 14 (2) (1990) 179–211.
[84] Y. Bengio, R. Ducharme, P. Vincent, C. Janvin, A neural probabilistic language model, J. Mach. Learn. Res. 3 (2003) 1137–1155.
[85] R. Collobert, J. Weston, A unified architecture for natural language processing: deep neural networks with multitask learning, in: Proceedings of ICML, Helsinki, Finland, 2008, pp. 160–167.
[86] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, CoRR, arXiv:1301.3781.
[87] G.A. Miller, W.G. Charles, Contextual correlates of semantic similarity, Lang. Cogn. Processes 6 (1) (1991) 1–28.
[88] S.P. Ponzetto, M. Strube, Knowledge derived from Wikipedia for computing semantic relatedness, J. Artif. Intell. Res. 30 (2007) 181–212.
[89] E. Yeh, D. Ramage, C.D. Manning, E. Agirre, A. Soroa, WikiWalk: random walks on Wikipedia for semantic relatedness, in: Proceedings of the 2009 Workshop on Graph-Based Methods for Natural Language Processing, Singapore, 2009, pp. 41–49.
[90] T. Zesch, C. Müller, I. Gurevych, Using Wiktionary for computing semantic relatedness, in: Proceedings of the 23rd National Conference on Artificial Intelligence, Chicago, Illinois, vol. 2, 2008, pp. 861–866.
[91] Z. Wu, C.L. Giles, Sense-aware semantic analysis: a multi-prototype word representation model using Wikipedia, in: Proceedings of the 29th AAAI Conference on Artificial Intelligence, Austin, TX, USA, 2015.
[92] D. Milne, I.H. Witten, Learning to link with Wikipedia, in: Proceedings of the 17th ACM Conference on Information and Knowledge Management, California, USA, 2008, pp. 509–518.
[93] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Paşca, A. Soroa, A study on similarity and relatedness using distributional and WordNet-based approaches, in: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Boulder, Colorado, 2009, pp. 19–27.
[94] T. Hughes, D. Ramage, Lexical semantic relatedness with random graph walks, in: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP–CoNLL, Prague, Czech Republic, 2007, pp. 581–589.
[95] E. Minkov, W.w. Cohen, Adaptive graph walk-based similarity measures for parsed text, Nat. Lang. Eng. 20 (2014) 361–397.
[96] R. Mihalcea, C. Corley, C. Strapparava, Corpus-based and knowledge-based measures of text semantic similarity, in: Proceedings of the 21st National Conference on Artificial Intelligence, AAAI'06, Boston, Massachusetts, vol. 1, 2006, pp. 775–780.
[97] L. Han, A.L. Kashyap, T. Finin, J. Mayfield, J. Weese, UMBC_EBIQUITY-CORE: semantic textual similarity systems, in: Proceedings of the Second Joint Conference on Lexical and Computational Semantics, Atlanta, Georgia, 2013, pp. 44–52.
[98] A.L. Kashyap, L. Han, R. Yus, J. Sleeman, T.W. Satyapanich, S.R. Gandhi, T. Finin, Meerkat Mafia: multilingual and cross-level semantic textual similarity systems, in: Proceedings of the 8th International Workshop on Semantic Evaluation, Dublin, Ireland, 2014, pp. 416–423.
[99] A.M. Sultan, S. Bethard, T. Sumner, Back to basics for monolingual alignment: exploiting word similarity and contextual evidence, Trans. Assoc. Comput. Linguist. 2 (1) (2014) 219–230.
[100] G. Tsatsaronis, I. Varlamis, M. Vazirgiannis, Text relatedness based on a word thesaurus, J. Artif. Intell. Res. 37 (1) (2010) 1–40.
[101] Y. Li, D. McLean, Z.A. Bandar, J.D. O'Shea, K. Crockett, Sentence similarity based on semantic nets and corpus statistics, IEEE Trans. Knowl. Data Eng. 18 (8) (2006) 1138–1150.
[102] P.D. Turney, Semantic composition and decomposition: from recognition to generation, Tech. rep., National Research Council of Canada, 2014.
[103] J. Mitchell, M. Lapata, Vector-based models of semantic composition, in: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, ACL, Columbus, Ohio, USA, 2008, pp. 236–244.
[104] J. Mitchell, M. Lapata, Composition in distributional models of semantics, Cogn. Sci. 34 (8).
[105] E. Grefenstette, M. Sadrzadeh, Concrete models and empirical evaluations for the categorical compositional distributional model of meaning, Comput. Linguist. 41 (1) (2015) 71–118.
[106] M. Franco-Salvador, P. Rosso, R. Navigli, A knowledge-based representation for cross-language document retrieval and categorization, in: Proceedings of the 14th Conference on European Chapter of the Association for Computational Linguistics, EACL, Gothenburg, Sweden, 2014.
[107] E. Agirre, D. Cer, M. Diab, A. Gonzalez-Agirre, SemEval-2012 task 6: a pilot on semantic textual similarity, in: Proceedings of SemEval-2012, Montreal, Canada, 2012, pp. 385–393.
[108] A. Islam, D. Inkpen, Semantic text similarity using corpus-based word similarity and string similarity, ACM Trans. Knowl. Discov. Data 2 (2) (2008) 10:1–10:25.
[109] L. Allison, T.I. Dix, A bit-string longest-common-subsequence algorithm, Inf. Process. Lett. 23 (6) (1986) 305–310.
[110] M.J. Wise, YAP3: improved detection of similarities in computer program and other texts, in: Proceedings of the Twenty-Seventh SIGCSE Technical Symposium on Computer Science Education, SIGCSE'96, Philadelphia, Pennsylvania, USA, 1996, pp. 130–134.
[111] D. Ramage, A.N. Rafferty, C.D. Manning, Random walks for text semantic similarity, in: Proceedings of the 2009 Workshop on Graph-Based Methods for Natural Language Processing, Suntec, Singapore, 2009, pp. 23–31.
[112] E.H. Hovy, R. Navigli, S.P. Ponzetto, Collaboratively built semi-structured content and Artificial Intelligence: the story so far, Artif. Intell. 194 (2013) 2–27.
[113] C.M. Meyer, I. Gurevych, To exhibit is not to loiter: a multilingual sense-disambiguated Wiktionary for measuring verb similarity, in: Proceedings of the 24th International Conference on Computational Linguistics, Mumbai, India, vol. 4, 2012, pp. 1763–1780.
[114] J. Ganitkevitch, B. Van Durme, C. Callison-Burch, PPDB: the paraphrase database, in: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta, Georgia, 2013, pp. 758–764.
[115] M. Baroni, A. Lenci, Distributional memory: a general framework for corpus-based semantics, Comput. Linguist. 36 (4) (2010) 673–721.
[116] E. Iosif, A. Potamianos, Similarity computation using semantic networks created from web-harvested data, Nat. Lang. Eng. 21 (2015) 49–79.
[117] M. Baroni, B. Murphy, E. Barbu, M. Poesio, Strudel: a corpus-based semantic model based on properties and types, Cogn. Sci. 34 (2) (2010) 222–254.
[118] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, J. Mach. Learn. Res. 12 (2011) 2493–2537.
[119] T.H. Haveliwala, Topic-sensitive PageRank, in: Proceedings of the 11th International Conference on World Wide Web, Hawaii, USA, 2002, pp. 517–526.
[120] S. Brin, M. Page, Anatomy of a large-scale hypertextual web search engine, in: Proceedings of the 7th Conference on World Wide Web, Brisbane, Australia, 1998, pp. 107–117.
[121] A. Montejo-Ráez, E. Martínez-Cámara, M.T. Martín-Valdivia, L.A. Ureña López, Random walk weighting over SentiWordNet for sentiment polarity detection on Twitter, in: Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis, Jeju, Republic of Korea, 2012, pp. 3–10.
[122] R. Mihalcea, P. Tarau, E. Figa, PageRank on semantic networks, with application to Word Sense Disambiguation, in: Proceedings of the 20th International Conference on Computational Linguistics, The 20th International Conference on Computational Linguistics, Geneva, Switzerland, 2004.

[123] E. Agirre, O. López de Lacalle, A. Soroa, Random walks for knowledge-based Word Sense Disambiguation, Comput. Linguist. 40 (1) (2014) 57–84.
[124] R. Navigli, S. Faralli, A. Soroa, O.L. de Lacalle, E. Agirre, 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 2011, 2011, pp. 2317–2320.
[125] A. Moro, A. Raganato, R. Navigli, Entity linking meets Word Sense Disambiguation: a unified approach, Trans. Assoc. Comput. Linguist. 2 (2014) 231–244.
[126] J. Wang, J. Liu, C. Wang, Keyword extraction based on PageRank, in: Proceedings of the 11th Pacific–Asia Conference on Advances in Knowledge Discovery and Data Mining, Nanjing, China, 2007, pp. 857–864.
[127] E. Niemann, I. Gurevych, The people's web meets linguistic knowledge: automatic sense alignment of Wikipedia and Wordnet, in: Proceedings of the Ninth International Conference on Computational Semantics, Oxford, United Kingdom, 2011, pp. 205–214.
[128] A.N. Langville, C.D. Meyer, Google's PageRank and Beyond: The Science of Search Engine Rankings, Princeton University Press, Princeton, NJ, USA, 2006.
[129] W. Webber, A. Moffat, J. Zobel, A similarity measure for indefinite rankings, ACM Trans. Inf. Syst. 28 (4) (2010) 20:1–20:38.
[130] M. Kendall, Rank Correlation Methods, Charles Griffin, London, 1948.
[131] A. Tversky, Features of similarity, Psychol. Rev. 84 (1977) 327–352.
[132] A. Markman, D. Gentner, Structural alignment during similarity comparisons, Cogn. Psychol. 25 (4) (1993) 431–467.
[133] W.A. Gale, K. Church, D. Yarowsky, One sense per discourse, in: Proceedings of DARPA Speech and Natural Language Workshop, Harriman, NY, USA, 1992, pp. 233–237.
[134] D. Jurgens, M.T. Pilehvar, R. Navigli, SemEval-2014 task 3: cross-level semantic similarity, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), in Conjunction with COLING 2014, Dublin, Ireland, 2014, pp. 17–26.
[135] T. Flati, D. Vannella, T. Pasini, R. Navigli, 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, 2014, pp. 945–955.
[136] T. Chklovski, R. Mihalcea, Building a sense tagged corpus with Open Mind Word Expert, in: Proceedings of the ACL-02 Workshop on WSD: Recent Successes and Future Directions at ACL-02, Philadelphia, PA, USA, 2002, pp. 116–122.
[137] B. Snyder, M. Palmer, The English all-words task, in: Proceedings of the 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, SENSEVAL-3, Barcelona, Spain, 2004, pp. 41–43.
[138] J. Gonzalo, F. Verdejo, I. Chugur, J. Cigarrán, Indexing with WordNet synsets can improve text retrieval, in: Proceedings of the COLING/ACL'98 Workshop on Usage of WordNet for NLP, Montréal, Canada, 1998, pp. 38–44.
[139] M. Carpuat, D. Wu, Improving statistical machine translation using Word Sense Disambiguation, in: The 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP–CoNLL 2007, Prague, Czech Republic, 2007, pp. 61–72.
[140] B. Dandala, C. Hokamp, R. Mihalcea, R.C. Bunescu, Sense clustering using Wikipedia, in: Proceedings of Recent Advances in Natural Language Processing, Hissar, Bulgaria, 2013, pp. 164–171.
[141] A. Kilgarriff, English lexical sample task description, in: Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems, SENSEVAL-2, Toulouse, France, 2001, pp. 17–20.
[142] R. Weischedel, S. Pradhan, L. Ramshaw, M. Palmer, N. Xue, M. Marcus, A. Taylor, C. Greenberg, E. Hovy, R. Belvin, A. Houston, OntoNotes release 2.0, ldc2008t04, CD-ROM, Linguistic Data Consortium, Philadelphia, Penn., 2008.
[143] S. Banerjee, T. Pedersen, 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, 2002, pp. 136–145.
[144] P.D. Turney, Mining the web for synonyms: PMI-IR versus LSA on TOEFL, in: Proceedings of the 12th European Conference on Machine Learning, Freiburg, Germany, 2001, pp. 491–502.
[145] C. Leacock, M. Chodorow, G. Miller, Using corpus statistics and WordNet relations for sense identification, Comput. Linguist. 24 (1) (1998) 147–166.
[146] D. Bär, T. Zesch, I. Gurevych, 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, 2013, pp. 121–126.
[147] T. Zesch, I. Gurevych, Analysis of the Wikipedia category graph for NLP applications, in: Proceedings of the Second Workshop on TextGraphs: Graph-Based Algorithms for Natural Language Processing, Rochester, NY, USA, 2007, pp. 1–8.
[148] T. Zesch, I. Gurevych, Wisdom of crowds versus wisdom of linguists – measuring the semantic relatedness of words, Nat. Lang. Eng. 16 (1) (2010) 25–59.
[149] F. Šarić, G. Glavaš, M. Karan, J. Šnajder, B. Dalbelo Bašić, TakeLab: systems for measuring semantic text similarity, in: Proceedings of SemEval-2012, Montreal, Canada, 2012, pp. 441–448.
[150] E. Agirre, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, *SEM 2013 shared task: semantic textual similarity, in: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, Association for Computational Linguistics, Atlanta, Georgia, USA, 2013, pp. 32–43.
[151] M. Heilman, N. Madnani, Henry-core: domain adaptation and stacking for text similarity, in: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, Atlanta, Georgia, USA, 2013, pp. 96–102.
[152] M. Diab, Semantic Textual Similarity: past, present and future, in: Proceedings of the Joint Symposium on Semantic Processing. Textual Inference and Structures in Corpora, 2013, p. 6, http://www.aclweb.org/anthology/W13-3806.
[153] E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, R. Weischedel, OntoNotes: the 90% solution, in: Proceedings of NAACL, NY, USA, 2006, pp. 57–60.