Natural Language Engineering (2019), **25**, pp. 693–714
doi:10.1017/S1351324919000305

CAMBRIDGE
UNIVERSITY PRESS

ARTICLE

An overview of word and sense similarity

Roberto Navigli and Federico Martelli*

Department of Computer Science, Sapienza University of Rome, Italy
*Corresponding author. Email: martelli@di.uniroma1.it

(Received 1 April 2019; revised 21 May 2019; first published online 25 July 2019)

Abstract

Over the last two decades, determining the similarity between words as well as between their meanings, that is, word senses, has been proven to be of vital importance in the field of Natural Language Processing. This paper provides the reader with an introduction to the tasks of computing word and sense similarity. These consist in computing the degree of semantic likeness between words and senses, respectively. First, we distinguish between two major approaches: the knowledge-based approaches and the distributional approaches. Second, we detail the representations and measures employed for computing similarity. We then illustrate the evaluation settings available in the literature and, finally, discuss suggestions for future research.

Keywords: semantic similarity; word similarity; sense similarity; distributional semantics; knowledge-based similarity

1. Introduction

Measuring the degree of semantic similarity between linguistic items has been a great challenge in the field of Natural Language Processing (NLP), a sub-field of Artificial Intelligence concerned with the handling of human language by computers. Over the last two decades, several different approaches have been put forward for computing similarity using a variety of methods and techniques. However, before examining such approaches, it is crucial to provide a definition of similarity: what is meant exactly by the term 'similar'? Are all semantically related items 'similar'? Resnik (1995) and Budanitsky and Hirst (2001) make a fundamental distinction between two apparently interchangeable concepts, that is, similarity and relatedness. In fact, while similarity refers to items which can be substituted in a given context (such as *cute* and *pretty*) without changing the underlying semantics, relatedness indicates items which have semantic correlations but are not substitutable. Relatedness encompasses a much larger set of semantic relations, ranging from antonymy (*beautiful* and *ugly*) to correlation (*beautiful* and *appeal*). As is apparent from Figure 1, *beautiful* and *appeal* are related but not similar, whereas *pretty* and *cute* are both related and similar. In fact, similarity is often considered to be a specific instance of relatedness (Jurafsky 2000), where the concepts evoked by the two words belong to the same ontological class. In this paper, relatedness will not be discussed and the focus will lie on similarity.

In general, semantic similarity can be classified on the basis of two fundamental aspects. The first concerns the type of resource employed, whether it be a lexical knowledge base (LKB), that is, a wide-coverage structured repository of linguistic data, or large collections of raw textual data, that is, corpora. Accordingly, we distinguish between *knowledge-based semantic similarity*, in the former case, and *distributional semantic similarity*, in the latter. Furthermore, *hybrid semantic similarity* combines both knowledge-based and distributional methods. The second aspect concerns the type of linguistic item to be analysed, which can be:

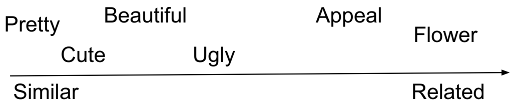

Figure 1. An explicative illustration of word similarity and relatedness.

- *words*, which are the basic building blocks of language, also including their inflectional information.
- *word senses*, that is, the meanings that words convey in given contexts (e.g., the device meaning vs. the animal meaning of *mouse*).
- *sentences*, that is, grammatical sequences of words which typically include a main clause, made up of a predicate, a subject and, possibly, other syntactic elements.
- *paragraphs* and *texts*, which are made up of sequences of sentences.

This paper focuses on the first two items, that is, words and senses, and provides a review of the approaches used for determining to which extent two or more words or senses are similar to each other, ranging from the earliest attempts to recent developments based on embedded representations.

1.1 Outline

The rest of this paper is structured as follows. First, we describe the tasks of word and sense similarity (Section 2). Subsequently, we detail the main approaches that can be employed for performing these tasks (Sections 3–5) and describe the main measures for comparing vector representations (Section 6). We then move on to the evaluation of word and sense similarity measures (Section 7). Finally, we draw conclusions and propose some suggestions for future research (Section 8).

2. Task description

Given two linguistic items i_1 and i_2, either words or senses in our case, the task consists in calculating some function $sim(i_1, i_2)$ which provides a numeric value that quantifies the estimated similarity between i_1 and i_2. More formally, the similarity function is of the kind:

$$sim : I \times I \longrightarrow \mathbb{R} \tag{1}$$

where I is the set of linguistic items of interest and the output of the function typically ranges between 0 and 1, or between -1 and 1. Note that the set of linguistic items can be cross-level, that is, it can include (and therefore enable the comparison of) items of different types, such as words and senses (Jurgens 2016).

In order to compute the degree of semantic similarity between items, two major steps have to be carried out. First, it is necessary to identify a suitable representation of the items to be analysed. The way a linguistic item is represented has a fundamental impact on the effectiveness of the computation of semantic similarity, as a consequence of the expressiveness of the representation. For example, a representation which counts the number of occurrences and co-occurrences of words can be useful when operating at the lexical level, but can lead to more difficult calculations when moving to the sense level, for example, due to the paucity of sense-tagged training data. Second, an effective similarity measure has to be selected, that is, a way to compare items on the basis of a specific representation.

Word and sense similarity can be performed following two main approaches:

- *Knowledge-based similarity* exploits explicit representations of meaning derived from wide-coverage lexical-semantic knowledge resources (introduced in Section 3).
- *Distributional similarity* draws on distributional semantics, also known as vector space semantics, and exploits the statistical distribution of words within unstructured text (introduced in Section 4).

Hybrid similarity measures, introduced in Section 5, combine knowledge-based and distributional similarity approaches, that is, knowledge from LKBs and occurrence information from texts.

3. Knowledge-based word and sense similarity

Knowledge-based approaches compute semantic similarity by exploiting the information stored in an LKB. With this aim in view, two main methods can be employed. The first method computes the semantic similarity between two given items i_1 and i_2 by inferring their semantic properties on the basis of structural information concerning i_1 and i_2 within a specific LKB. The second method performs the extraction and comparison of a vector representation of i_1 and i_2 obtained from the LKB. It is important to note that the first method is now deprecated as the best performance can be achieved by using more sophisticated techniques, both knowledge-based and distributional, which we will detail in the following sections.

We now introduce the most common LKBs (Section 3.1), and then overview methods and measures employed for knowledge-based word and sense similarity (Section 3.2).

3.1 Lexical knowledge resources

Here we will review the most popular lexical knowledge resources, which are widely used not only for computing semantic similarity, but also in many other NLP tasks.

WordNet. WordNet[a] (Fellbaum 1998) is undoubtedly the most popular LKB for the English language, originally developed on the basis of psycholinguistic theories. WordNet can be viewed as a graph, whose nodes are synsets, that is, sets of synonyms, and whose edges are semantic relations between synsets. WordNet encodes the meanings of an ambiguous word through the synsets which contain that word and therefore the corresponding senses. For instance, for the word *table*, WordNet provides the following synsets, together with a textual definition (called gloss) and, possibly, usage examples:

- { table, tabular array } – a set of data arranged in rows and columns '*see table 1*'.
- { table } – a piece of furniture having a smooth flat top that is usually supported by one or more vertical legs '*it was a sturdy table*'.
- { table } – a piece of furniture with tableware for a meal laid out on it '*I reserved a table at my favorite restaurant*'.
- { mesa, table } – flat tableland with steep edges '*the tribe was relatively safe on the mesa but they had to descend into the valley for water*'.
- { table } – a company of people assembled at a table for a meal or game '*he entertained the whole table with his witty remarks*'.
- { board, table } – food or meals in general '*she sets a fine table*'; '*room and board*'.

In the above example, the term *tabular array* is a synonym for *table* in the data matrix sense, while *mesa* is a synonym in the tableland meaning. WordNet makes clear the important distinction between words, senses and synsets: a word is a possibly ambiguous string which represents a single, meaningful linguistic element (e.g., table), a sense is a given meaning of a certain word (e.g., the matrix sense of table, also denoted as table#n#1, table.n.1 or table_n^1 to indicate it is the first nominal sense in the WordNet inventory for that word) and a synset is a set of senses all expressing the same concept. A synset has a one-to-one correspondence with a concept, which is purely semantic. A sense (e.g., table_n^1) uniquely identifies the only synset it occurs in (e.g., table_n^1 identifies the 08283156n id in WordNet 3.1 of the synset { table, tabular array }), whereas given a synset S and a word $w \in S$ (e.g., the word table in the { table, tabular array } synset), a sense s of w is uniquely identified (i.e., table_n^1 in the example).

[a] http://wordnetweb.princeton.edu.

Synsets are connected via different kinds of relations, the most popular being:

- *Hypernymy* (and its inverse *hyponymy*), which denotes generalization: for instance, the { table, tabular array } is a kind of { array }, while { table }, in the furniture meaning, is a kind of { furniture, piece of furniture, article of furniture }.
- *Meronymy* (and its inverse *holonymy*), which denotes a part of relationship: for instance, { table, tabular array } has part { row } and { column }, whereas { table } in the furniture meaning has part, among others, { leg }.
- *Similarity*, which specifies the similarity between adjectival synsets such as between { beautiful } and { pretty }.
- *Pertainymy* and *derivationally related form*, which connect word senses from different parts of speech with a common root stem, such as that relating { table, tabular array } to the { tabulate } verbal synset, and the same nominal synset to the { tabular } adjectival synset.

Note that some of the above relations are semantic, in that they connect synsets, whereas others, such as the derivationally related form and the pertainymy relations, hold between word senses (i.e., words occurring in synsets). However, it is a common practice, for the purposes of many NLP tasks, to take lexical relations to the semantic level, so as to connect the corresponding enclosing synsets (Navigli and Ponzetto 2012).

Roget's thesaurus. Created by the English lexicographer Peter Mark Roget in 1805, the Roget's thesaurus is a historical lexicographic resource, used in NLP as an alternative to WordNet for knowledge acquisition and semantic similarity (Jarmasz and Szpakowicz 2003). The Roget's thesaurus was made available for the first time in 1852 and was one of the resources employed for creating WordNet (Miller *et al.* 1990).

Wikipedia. Started in 2001, Wikipedia[b] has become the largest and most reliable online encyclopaedia in the space of a few years and has gained momentum quickly in several NLP tasks, such as text classification (Navigli *et al.* 2011), Word Sense Disambiguation (WSD) (Navigli, Jurgens, and Vannella 2013; Moro and Navigli 2013), entity linking (Moro and Navigli 2013) and many others (Hovy, Navigli, and Ponzetto 2013). Wikipedia can be viewed as a lexical knowledge resource with a graph structure whose nodes are Wikipedia pages and whose relations are given by the hyperlinks that connect one page to another. Compared to WordNet, Wikipedia provides three key features which make it very popular in NLP: first, it covers world knowledge in terms of named entities (such as well-known people, companies and works of art) on a large scale; second, it provides coverage of multiple languages, by linking a given page to its counterparts in dozens of other languages, whenever these are available; and third, it is continuously updated.

Wiktionary. Another resource which has become popular in NLP is Wiktionary, a sister project to Wikipedia. Available in almost 200 languages, Wiktionary is a free, Web-based collaborative dictionary that is widely employed in several NLP tasks such as WSD and semantic similarity (Zesch, Müller, and Gurevych 2008).

BabelNet. Built on top of WordNet and Wikipedia, BabelNet[c] (Navigli and Ponzetto 2012) is the most popular wide-coverage multilingual lexical knowledge resource, used in dozens of tasks among which we cite state-of-the-art multilingual disambiguation (Moro, Raganato, and Navigli 2014), semantic similarity (Camacho-Collados, Pilehvar, and Navigli 2016) and semantically enhanced machine translation (Moussallem, Wauer, and Ngonga Ngomo 2018).

BabelNet is the result of the automatic interlinking and integration of different knowledge resources, such as WordNet, Wikipedia, Wiktionary, Wikidata and other resources. The underlying structure is modelled after that of WordNet: multilingual synsets are created which contain

[b] https://www.wikipedia.org.
[c] https://babelnet.org.

lexicalizations that, in different languages, express the same concept. For instance, the *car* synset includes, among others, the following lexicalizations (the language code is subscripted):

$$\{ \text{car}_{EN}, \text{automobile}_{EN}, \text{macchina}_{IT}, \text{voiture}_{FR}, \text{coche}_{ES}, ..., \text{Wagen}_{DE} \}.$$

The relations interconnecting the BabelNet multilingual synsets come from the integrated resources, such as those from WordNet and Wikipedia (where hyperlinks are labelled as semantic relatedness relations). As a result, similar to WordNet and Wikipedia, BabelNet can also be viewed as a graph and its structure exploited to perform semantic similarity.

3.2 Knowledge-based representations and measures

3.2.1 Earlier attempts

Knowledge-based representations and measures always rely on the availability of LKBs. Earlier efforts aimed at calculating word and sense similarity by exploiting solely the taxonomic structure of an LKB, such as WordNet. The structural information usually exploited by these measures is based on the following ingredients:

- The *depth* of a given concept (i.e., synset) in the LKB taxonomy;
- The *length* of the shortest path between two concepts in the LKB;
- The *Least Common Subsumer (LCS)*, that is, the lowest concept in the taxonomical hierarchy which is a common hypernym of two target concepts.

In knowledge-based approaches, computing the similarity between two senses s_1 and s_2 is straightforward, because it involves the calculation of a measure concerning the two corresponding nodes in the LKB graph. When two words w_1 and w_2 are involved, instead, the similarity between them can be computed as the maximum similarity across all their sense combinations:

$$sim(w_1, w_2) = \max_{s_1 \in Senses(w_1),\, s_2 \in Senses(w_2)} sim(s_1, s_2) \tag{2}$$

where $Senses(w_i)$ is the set of senses provided in the LKB for word w_i.

Path. One of the earliest and simplest knowledge-based algorithms for the computation of semantic similarity is based on the assumption that the shorter the path in a specific LKB graph between two senses, the more semantically similar they are. Given two senses s_1 and s_2, the path length (Rada *et al.* 1989) can be computed as follows:

$$sim(s_1, s_2) = \frac{1}{length(s_1, s_2) + 1} \tag{3}$$

where we adjusted the original formula to a similarity measure by calculating its reciprocal. Related to this approach, but based on the structural distance within the Roget's thesaurus (see Section 3.1), a similar algorithm has been put forward by Jarmasz and Szpakowicz (2003).

The key idea behind this type of algorithms is that the farther apart the senses of the two words of interest in the LKB are, the lower the degree of similarity between the two words is.

Leacock and Chodorow. A variant of the path measure was proposed by Leacock (1998), and this computes semantic similarity as:

$$sim(s_1, s_2) = -log \frac{length(s_1, s_2)}{2D} \tag{4}$$

where *length* refers to the shortest path between the two senses and D is the maximum depth of the (nominal) taxonomy of a given LKB (historically, WordNet).

Wu and Palmer. In order to take into account the taxonomical information shared by two senses, Wu and Palmer (1994) put forward the following measure:

$$sim(s_1, s_2) = \frac{2 \cdot depth(LCS(s_1, s_2))}{depth(s_1) + depth(s_2)} \tag{5}$$

where the higher the LCS, the lower the similarity between s_1 and s_2.

Resnik. A more sophisticated approach was proposed by Resnik (1995) who developed a notion of information content which determines the amount of information covered by a certain WordNet synset in terms of all its descendants (i.e., hyponyms). Formally, this similarity measure is computed as follows:

$$sim(s_1, s_2) = IC(LCS(s_1, s_2)) \tag{6}$$

where *IC*, that is, the information content, is defined as:

$$IC(S) = -logP(S) \tag{7}$$

where $P(S)$ is the probability that a word, randomly selected within a large corpus, is an instance of a given synset S. Such probability is calculated as:

$$P(S) = \frac{\sum_{w \in words(S)} count(w)}{N} \tag{8}$$

where $words(S)$ is the set of words contained in synset S and all its hyponyms, $count(w)$ is the number of occurrences of w in the reference corpus and N is the total number of word tokens in the corpus.

Lin. A refined version of Resnik's measure was put forward by Lin (1998) which exploits the information content not only of the commonalities, but also of the two senses individually. Formally:

$$sim(s_1, s_2) = \frac{2 \cdot IC(LCS(s_1, s_2))}{IC(s_1) + IC(s_2)} \tag{9}$$

Jiang and Conrath. A variant of Lin's measure that has been widely used in the literature is the following (Jiang and Conrath 1997):

$$sim(s_1, s_2) = \frac{1}{IC(s_1) + IC(s_2) - 2 \cdot IC(LCS(s_1, s_2))} \tag{10}$$

Extended gloss overlaps or Extended Lesk. All of the above approaches are hinged on taxonomic information, which however is only a portion of the information that is provided in LKBs such as WordNet. Other kinds of relations can indeed be used, such as meronymy and pertainymy (cf. Section 3.1). To do this, Banerjee and Pedersen (2003) proposed an improvement of the Lesk algorithm (Lesk 1986), which has been used historically in WSD for determining the overlap between the textual definitions of two senses under comparison. The measure designed by Banerjee and Pedersen (2003) extends this idea by considering the overlap between definitions not only of the target senses, but also of their neighbouring synsets in the WordNet graph:

$$sim(s_1, s_2) = \sum_{r, r' \in R} \sum_{s \in r(s_1)} \sum_{s' \in r'(s_2)} overlap(gloss(s), gloss(s')) \tag{11}$$

where R is the set of lexical-semantic relations in WordNet, *gloss* is a function that provides the textual definition for a given synset (or sense), *overlap* determines the number of common words between two definitions and $r(s)$ provides the set of the other endpoints of the relation edges of type r connecting s.

Wikipedia-based semantic relatedness. One of the advantages of using Wikipedia as opposed to WordNet is the former's network of interlinked articles. A key hunch is that two articles are deemed similar if they are linked by a similar set of pages (Milne and Witten 2008). Such similarity can be computed with the following formula:

$$sim(a, b) = 1 - \frac{log(max(|in(a)|, |in(b)|)) - log(|in(a) \cap in(b)|)}{log(|W|) - log(min(|in(a)|, |in(b)|))} \tag{12}$$

where a and b are two Wikipedia articles, $in(a)$ is the set of articles linking to a, and W is the full set of Wikipedia articles. This measure aims at determining the degree of relatedness between two articles, nonetheless when the two articles are close enough (i.e., the value gets close to 1) the computed value can be considered a degree of similarity.

3.2.2 Recent developments

More recent knowledge-based approaches extract vector-based representations of meaning, which are then used to determine semantic similarity. Unlike previous techniques where the main form of linguistic knowledge representation was the LKB itself, in this case a second form of linguistic knowledge representation is involved, namely, a vector encoding. Accordingly, word and sense similarity is computed in two steps:

- a vector-based word and sense representation is obtained by exploiting the structural information of an LKB.
- the obtained vector representations are compared by applying a similarity measure.

In what follows we overview approaches to the first step, while deferring an introduction to similarity measures to Section 6.

Personalized PageRank-based representations. A key idea introduced in the scientific literature is the exploitation of Markov chains and random walks to determine the importance of nodes in the graph, and this was popularized with the PageRank algorithm (Page *et al.* 1998). In order to obtain probability distributions specific to a node, that is, a concept of interest, topic-sensitive or Personalized PageRank (PPR) (Haveliwala 2002) is employed for the calculation of a semantic signature for each WordNet synset (Pilehvar, Jurgens and Navigli 2013).

Given the WordNet adjacency matrix M (possibly enriched with further edges, for example, from disambiguated WordNet glosses), the following formula is computed:

$$v_t = \alpha M v_{t-1} + (1 - \alpha) v_0 \tag{13}$$

where v_0 denotes the probability distribution for restart of the random walker in the network and α is the so-called damping factor (typically set to 0.85). The result of the computation of the above PageRank formula in the topic-sensitive setting (i.e., when v_0 is highly skewed) provides a distribution with most of the probability mass concentrated on the nodes, which are at easy reach from the nodes initialized for restart in v_0. Depending on how v_0 is initialized, an explicit semantic representation for a target word or sense can be obtained. For the target word w, it is sufficient to initialize the components of v_0 corresponding to the senses of w to $1/|Senses(w)|$ (i.e., uniformly distributed across the synsets of w in WordNet), and 0 for all other synsets. For computing a representation of a target sense s of a word w, v_0 is, instead, initialized to 1 on the corresponding synset, and 0 otherwise. An alternative approach has been proposed (Hughes and Ramage 2007) which interconnects not only synsets, but also words and POS-tagged words. Some variants also link synset and words in their definition, and use sense-occurrence frequencies to weight edges. However, this approach is surpassed in performance by purely synset-based semantic signatures when using a suitable similarity measure (Pilehvar *et al.* 2013).

4. Distributional word and sense similarity

Knowledge-based approaches can only be implemented if a lexical-semantic resource such as WordNet or BabelNet is available. A radically different approach which does not rely on structured knowledge bases exploits the statistical distribution of words occurring in corpora. The fundamental assumption behind distributional approaches is that the semantic properties of a given word *w* can be inferred from the contexts in which *w* appears. That is, the semantics of *w* is determined by all the other words which co-occur with it (Harris 1954; Firth 1957).

4.1 Corpora

Distributional approaches rely heavily on corpora, that is, large collections of raw textual data which can be leveraged effectively for computing semantic similarity. In fact, large-scale corpora reflect the behaviour of words in context, that is, they reveal a wide range of relationships between words, making them a particularly suitable resource from which to learn word distributions. These are then used to infer semantic properties and determine the extent of semantic similarity between two words.

The most widely employed corpora for word and sense similarity are:

- *Wikipedia*, one of the largest multilingual corpora employed in several NLP tasks.
- *UMBC*[d] (Han and Finin 2013), a Web corpus including more than three billion English words derived from the Stanford WebBase project.
- *ukWaC*[e] (Ferraresi *et al.* 2008), a 2-billion word corpus constructed using the .uk domain and medium-frequency words from the British National Corpus.
- *GigaWord*[f] (Graff *et al.* 2003), a large corpus of newswire text that has been acquired over several years by the Linguistic Data Consortium (LDC).

4.2 Distributional representations and measures

In the distributional approach, a vector representation typically encodes the behavioural use of specific words and/or senses. Two types of distributional representation can be distinguished:

- *Explicit representation*, which refers to a form of representation in which every dimension can be interpreted directly (e.g., when words or senses are used as the meanings of the vector's dimensions).
- *Implicit or latent representation*, which encodes the linguistic information in a form which cannot be interpreted directly.

In the case of an explicit representation vector, given a word *w* and a vocabulary of size *N*, a feature vector specifies whether each vocabulary entry, that is, each word w', occurs in the neighbourhood of *w*. The size of a feature vector can range from the entire vocabulary size, that is, *N*, to two dimensions referring to the words preceding and succeeding the target word *w*. In many cases, most frequent words, such as articles, are not included in feature vectors as they do not contain useful semantic information regarding a particular word. Given the feature vector of the target word *w*, its dimensions can be:

- *binary values*, that is, 0 or 1 depending on whether a specific word co-occurs with the target word or not.
- *association and probabilistic measures* which provide the score or probability that a specific word co-occurs with the target word.

[d] https://ebiquity.umbc.edu/resource/html/id/351.

[e] http://wacky.sslmit.unibo.it/doku.php?id=corpora.

[f] https://catalog.ldc.upenn.edu/LDC2003T05.

A typical example of a binary-valued explicit vector representation is the so-called one-hot encoding of a word w, which is a unit vector $(0, 0, \ldots, 0, 1, 0, \ldots, 0)$, where only the dimension corresponding to word w is valued with 1. Latent representations, such as embeddings, instead, encode features which are not human-readable and are not directly associated with linguistic items.

In the rest of this section we introduce the two types of representations.

4.2.1 Explicit representations

Early distributional approaches aimed at capturing semantic properties directly between words depending on their distributions. To this end, different measures were proposed in the literature.

Sørensen-Dice index, also known as Dice's coefficient, is used to measure the similarity of two words. Formally:

$$sim(w_1, w_2) = \frac{2w_{1,2}}{w_1 + w_2} \tag{14}$$

where w_i is the number of occurrences of the corresponding word and $w_{1,2}$ is the number of co-occurrences of w_1 and w_2 in the same context (e.g., sentence).

Jaccard Index. Also known as Jaccard similarity coefficient, Jaccard Index (JI) is defined as follows:

$$sim(w_1, w_2) = \frac{w_{1,2}}{w_1 + w_2 - w_{1,2}} \tag{15}$$

which has clear similarities to the Sørensen-Dice index defined above. This measure was used for detecting word and sense similarity by Grefenstette (1994).

Pointwise Mutual Information. Given two words, the Pointwise Mutual Information (PMI) quantifies the discrepancy between their joint distribution and their individual distributions, assuming independence:

$$PMI(w_1, w_2) = log\frac{p(w_1, w_2)}{p(w_1)p(w_2)} = log\frac{D \cdot c(w_1, w_2)}{c(w_1)c(w_2)} \tag{16}$$

where w_1 and w_2 are two words, $c(w_i)$ is the count of w_i, $c(w_1, w_2)$ is the number of times the two words co-occur in a context and D is the number of contexts considered. This measure was introduced into NLP by Church and Hanks (1990).

Positive PMI. Because many entries of word pairs are never observed in a corpus, and therefore have their PMI equal to $log\ 0 = -\infty$, a frequently used version of PMI is one in which all negative values are flattened to zero:

$$PPMI(w_1, w_2) = \begin{cases} PMI(w_1, w_2) & \text{if } PMI(w_1, w_2) > 0 \\ 0 & \text{otherwise} \end{cases}$$

PPMI is among the most popular distributional similarity measures in the NLP literature.

The above association measures can be used to populate an explicit representation vector in which each component values the correlation strength between the word represented by the vector and the word identified by the component.

We now overview an approach that uses concepts as a word's vector components.

Explicit Semantic Analysis. An effective method, called Explicit Semantic Analysis (ESA), encodes semantic information in the word vector's components starting from Wikipedia (Gabrilovich and Markovitch 2007). The dimensionality of the vector space is given by the set of Wikipedia pages and the vector v_w for a given word w is computed by setting its i-th component $v_{w,i}$ to the TF-IDF of w in the i-th Wikipedia page p_i. Formally:

$$v_{w,i} = \text{TF-IDF}(w, p_i) = tf(w, p_i) \cdot log\frac{N}{N_w} \tag{17}$$

where p_i is the i-th page in Wikipedia, $tf(w, p_i)$ is the frequency of w in page p_i, N is the total number of Wikipedia pages and N_w is the number of Wikipedia pages in which w occurs.

It has been shown that using Wiktionary instead of Wikipedia leads to higher results in semantic similarity and relatedness (Zesch *et al.* 2008).

4.2.2 Implicit or latent representations

Latent Semantic Analysis. Latent Semantic Analysis (LSA) (Deerwester *et al.* 1990) is a technique used for inferring semantic properties of linguistic items starting from a corpus. A term-passage matrix is created whose rows correspond to words and whose columns correspond to passages in the corpus where words occur. At the core of LSA lies the singular value decomposition of the term-passage matrix which decomposes it into the product of three matrices. The dimensionality of the decomposed matrices is then reduced. As a result, latent representations of terms and passages are produced and comparisons between terms can be performed by just considering the rows of the lower-ranking term-latent dimension matrix.

Word embeddings. In the last few years, LSA has been superseded by neural approaches aimed at learning latent representations of words, called word embeddings. Different embedding techniques have been developed and refined. Earliest approaches representing words by means of continuous vectors date back to the late 1980s (Rumelhart, Hinton, and Williams 1988). A well-known technique which aims at acquiring distributed real-valued vectors as a result of learning a language model was put forward by Bengio (2003). Collobert (2011) proposed a unified neural network architecture for various NLP tasks. More recently, Mikolov (2013) proposed a simple technique which speeds up the learning and has proven to be very effective.

Word2vec. Undoubtedly, the most popular yet simple approach to learning word embeddings is called word2vec (Mikolov *et al.* 2013).[g] As in all distributional methods, the assumption behind word2vec is that the meaning of a word can be inferred effectively using the context in which that word occurs. Word2vec is based on a two-layer neural network which takes a corpus as input and learns vector-based representations for each word. Two variants of word2vec have been put forward:

(1) continuous bag of words (CBOW), which exploits the context to predict a target word;
(2) skip-gram, which, instead, uses a word to predict a target context word.

Focusing on the skip-gram approach, for each given target word w_t, the objective function of the neural network is set to maximize the conditional probabilities of the words surrounding w_t in a window of m words. Formally, the following log likelihood is calculated:

$$J(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{-m \le j \le m\,:\, j \ne 0} log(p(w_{t+j}|w_t)) \tag{18}$$

where T is the number of words in the training corpus and θ are the embedding parameters. Word2vec can be viewed as a close approximation of traditional window-based distributional approaches.

A crucial feature of word2vec consists in preserving relationships between vectors such as analogy. For instance, London - UK + Italy should be very close to Rome. Standard word2vec embeddings are available in English, obtained from the Google News dataset. Embeddings for dozens of languages can also be derived from Wikipedia or other corpora.

[g] https://code.google.com/archive/p/word2vec/.

fastText. Recently, an extension of word2vec's skip-gram model called fastText[h] has been proposed which integrates subword information (Joulin *et al.* 2017). A key difference between fastText and word2vec is that the former is capable of building vectors for misspellings or out-of-vocabulary words. This is taken into account thanks to encoding words as a bag of character n-grams (i.e., substrings of length n) together with the word itself. For instance, to encode the word table, the following 3-grams are considered: { `<ta, tab, abl, ble, le>` } ∪ { `<table>` }. Thanks to this, compared to the standard skip-gram model, the input vocabulary includes the word and all the n-grams that can be calculated from it. As a result, the meanings of prefixes and suffixes are also considered in the final representation, therefore reducing data sparsity. fastText provides ready sets of embeddings for around 300 languages, which makes it appealing for multilingual processing of text.[i]

GloVe. A key difference between LSA and word2vec is that the latter produces latent representations with the useful property of preserving analogies, therefore indicating appealing linear substructures of the word vector space, whereas the former takes better advantage of the overall statistical information present in the input documents. However, the advantage of either approach is the drawback of the other. GloVe (Global Vectors)[j] addresses this issue by performing unsupervised learning of latent word vector representations starting from global word–word co-occurrence information (Pennington, Socher, and Manning 2014). At the core of the approach is a counting method which calculates a co-occurrence vector $count_i$ for a word w_i with its j-th component counting the times w_j co-occurs with w_i within a context window of a certain size, where each individual count is weighted by $1/d$, and d is the distance between the two words in the context under consideration. The key constraint put forward by GloVe is that, for any two words w_i and w_j:

$$v_i^T v_j + b_i + b_j = log(count_{i,j}) \tag{19}$$

where v_i and v_j are the embeddings to learn and b_i and b_j are scalar biases for the two words. A least square regression model is then calculated which aims at learning latent word vector representations such that the loss function is driven by the above soft constraint for any pair of words in the vocabulary:

$$J = \sum_{i=1}^{V} \sum_{j=1}^{V} f(count_{i,j})(w_i^T w_j + b_i + b_j - log(count_{i,j}))^2 \tag{20}$$

where $f(count_{i,j})$ is a weighting function that reduces the importance of overly frequent word pairs and V is the size of the vocabulary. While both word2vec and GloVe are popular approaches, a key difference between the two is that the former is a predictive model, whereas the latter is a count-based model. While it has been found that standard count-based models such as LSA fare worse than word2vec (Mikolov *et al.* 2013; Baroni, Dinu, and Kruszewski 2014), Levy and Goldberg (2014) showed that a predictive model such as word2vec's skip-gram is essentially a factorization of a variant of the PMI co-occurrence matrix of the vocabulary, which is count-based. Experimentally, there is contrasting evidence as to the superiority of word2vec over GloVe, with varying (and, in several cases, not very different) results.

SensEmbed. Word embeddings conflate different meanings of a word into a single vector-based representation and are therefore unable to capture polysemy. In order to address this issue, Iacobacci *et al.* (2015) proposed an approach for obtaining embedded representations of word senses called sense embeddings. To this end, first, a text corpus is disambiguated with a state-of-the-art WSD system, that is, Babelfy (Moro *et al.* 2014); second, the disambiguated corpus

[h] `https://fasttext.cc/`.

[i] `https://fasttext.cc/docs/en/pretrained-vectors.html`.

[j] `https://nlp.stanford.edu/projects/glove/`.

is processed with word2vec, in particular with the CBOW architecture, in order to produce embeddings for each sense of interest.

AutoExtend. An alternative approach put forward by Rothe and Schütze (1995) takes into account the interactions and constraints between words, senses and synsets, as made available in the WordNet LKB, and – starting from arbitrary word embeddings – acquires the latent, embedded representations of senses and synsets by means of an autoencoder neural architecture, where word embeddings are at the input and output layers and the hidden layer provides the synset embeddings.

Contextual Word Embeddings. Recent approaches exploit the distribution of words to learn latent encodings which represent word occurrences in a given context. Two prominent examples of such approaches are ELMo (Peters *et al.* 2018) and BERT (Devlin *et al.* 2018). These approaches can be employed for tasks such as question answering, textual entailment and semantic role labelling, but also in tasks such as Word in Context (WiC) similarity (Pilehvar and Camacho-Collados 2019). While such approaches could potentially be used to produce word representations based on the vectors output at their first layer, their main goal is to work on contextualized linguistic items.

5. Hybrid word and sense similarity

Recently, some approaches have been put forward which bring together knowledge-based and distributional similarity by combining the knowledge provided in LKBs and the occurrence information obtained from texts. The key advantage of such approaches is their ability to embed words and meanings in the same vector space model.

Salient Semantic Analysis. A development along the lines of Explicit Semantic Analysis (cf. Section 4.2.1) has been put forward which exploits the hyperlink information in Wikipedia pages to determine the salience of concepts (Hassan and Mihalcea 2011). Specifically, given a Wikipedia page and a hyperlink, all the occurrences of its surface form (i.e., the hyperlinked text) are searched across the page and the sense annotations propagated to those occurrences. Additional non-linked phrases in the page are tagged with a Wikification system. A semantic profile for each page is then created by building a PMI vector of the co-occurrences of each term with the concept of interest in the entire Wikipedia corpus. Since Salient Semantic Analysis relies not only on the distribution of words occurring in Wikipedia pages, but also on the usage of the Wikipedia sense inventory and the manual linking of salient concepts to Wikipedia pages, this technique can be considered both distributional and knowledge-based.

Novel Approach to a Semantically-Aware Representation of Items. As described previously, a knowledge-based approach such as PPR obtains a vector representation for each synset in a WordNet-like graph, which contains semantic information consisting mostly of a prescribed nature, that is, the kind of information that can be found in a dictionary. Instead, to create vectors that account for descriptive information about the concept of interest, a Novel Approach to a Semantically-Aware Representation of Items (NASARI) has been put forward by Camacho-Collados *et al.* (2016). This approach exploits the distributional semantics of texts which describe the concept. For this purpose, given a target concept $c = (p, S)$ identified by the pair of Wikipedia page p and WordNet synset S which are linked in BabelNet (cf. Section 3.1), two steps are carried out. First, the contextual information for c is collected according to the following equation:

$$\mathcal{T}_\mathrm{c} = \mathcal{L}_\mathrm{p} \cup \mathcal{B}(\mathcal{R}_\mathrm{S}) \tag{21}$$

where $\mathcal{T}_\mathrm{c}$ is the contextual information for a specific concept $c = (p, S)$, $\mathcal{L}_\mathrm{p}$ is the set of Wikipedia pages containing the page p and all the pages pointing to p, $\mathcal{B}$ is a function which maps each WordNet synset S to the corresponding Wikipedia page p, $\mathcal{R}_\mathrm{S}$ is a set of synsets which contains S and all its related, that is, connected, synsets. Second, the contextual information $\mathcal{T}_c$ is encoded in a lexical vector representation. After tokenization, lemmatization and stopword removal, a bag of words is created from the Wikipedia pages in $\mathcal{L}_\mathrm{p}$. Lexical specificity is used for the identification of the most characteristic words in the bag of words extracted (Lafon 1980). As a result, a lexical vector is created which represents c.

In order to overcome the sparseness which can occur in such lexical vector representation, two additional versions of NASARI vectors are provided:

- **semantic vectors:** a synset-based representation is created whose dimensions are not potentially ambiguous words, but concepts (represented by synsets): for each word w in the bag of words of concept c, the set of all hypernyms in common between pairs of words in the vector is considered (e.g., table, lamp and seat are grouped under the hypernym { furniture, piece of furniture, article of furniture }) and encoded in a single dimension represented by that common hypernym. Finally, lexical specificity is calculated in order to determine the most relevant hypernyms that have to be encoded in the vector representation. The weight associated with each semantic dimension is determined by computing lexical specificity on the words grouped by each hypernym as if they were a single word in the underlying texts.
- **embedded vectors:** an alternative representation, which is latent and compact, rather than explicit like the semantic vectors, is also provided in the NASARI framework. Starting from the lexical vector $v_{lex}(S)$ of a synset S, the embedded vector $e(S)$ of S is computed as the weighted average of the embeddings of the words in the lexical vector. Formally:

$$e(s) = \frac{\sum_{w \in v_{lex}(S)} rank(w, v_{lex}(S))^{-1} e(w)}{\sum_{w \in v_{lex}(S)} rank(w, v_{lex}(S))^{-1}} \tag{22}$$

  where $e(w)$ is the word embedding (e.g., a word2vec embedding) of w and $rank(w, v_{lex}(S))$ is the ranking of word w in the sorted vector $v_{lex}(S)$.

DeConf. The main idea behind de-conflated semantic representations (DeConf) (Pilehvar and Collier 2016) is to develop a method for obtaining a semantic representation which embeds word senses into the same semantic space of words, analogously to the NASARI embedded vectors. At the core of this approach lies the computation of a list of sense-biasing words for each word sense of interest which is used to 'bend' the sense representations in the right direction. Specifically, DeConf is computed in two steps:

- *identification of sense-biasing words*, that is, a list of words is extracted from WordNet which most effectively represent the semantics of a given synset S. Such list is obtained by, first, applying the PPR algorithm to the WordNet graph with restart on S and, second, progressively adding new words from the WordNet synsets sorted in descending order by their PPR probability.
- *learning a sense representation*, calculated for a target sense s of a word w as follows:

$$v_s = \frac{\alpha v_w + \sum_{b \in B_s} \delta_s(b) v_b}{\alpha + \sum_{b \in B_s} \delta_s(b)} \tag{23}$$

  whose numerator is an average of the word embedding v_w weighted with a factor α and the embeddings of the various words in the list B_s of sense-biasing words calculated as a result of the first step, and weighted with a function $\delta_s(b)$ of their ranking in the list.

6. Measures for comparing vector-based representations

In this section, we focus on the main measures which are employed widely whenever two or more vector-based representations have to be compared for determining the degree of semantic similarity. The following measures are used in knowledge-based and distributional approaches, whenever they resort to vector-based representations.

Cosine similarity. Widely used for determining the similarity between two vectors, the cosine similarity is formalized as follows:

$$sim(w_1, w_2) = \frac{w_1 \cdot w_2}{\|w_1\| \, \|w_2\|} = \frac{\sum_{i=1}^{n} w_{1\mathrm{i}} w_{2\mathrm{i}}}{\sqrt{\sum_{i=1}^{n} {w_{1\mathrm{i}}}^2} \sqrt{\sum_{i=1}^{n} {w_{2\mathrm{i}}}^2}} \tag{24}$$

where w_1 and w_2 are two vectors to be compared. The above formula determines the closeness of the two vectors by calculating the dot product between them divided by their norms.

Weighted overlap. This measure (Camacho-Collados *et al.* 2016) compares the similarity between vectors in two steps. First, a ranking of the dimensions of each vector is calculated. Such ranking considers only dimensions with values different from 0 for both vectors, assigning higher scores to more relevant dimensions. Second, it sums the ranking of the two vectors normalized by a factor which computes the best rank pairing. The weighted overlap is formalized as follows:

$$WO(v_1, v_2) = \frac{\sum_{q \in O} (r_q^1 + r_q^2)^{-1}}{\sum_{i=1}^{|O|} (2i)^{-1}} \tag{25}$$

where O indicates the set of overlapping dimensions and r_q^i refers to the ranking of the q-th dimension of vector v_i ($i \in \{1, 2\}$). While cosine similarity is applicable to both latent and explicit vectors, Weighted Overlap is suitable only for explicit and potentially sparse vectors which have human-interpretable components, like in the knowledge-based vector representations produced with PPR or NASARI (see Sections 3.2.2 and 5).

7. Evaluation

We now describe how to evaluate and compare measures for word and sense similarity. We distinguish between:

- *in vitro* or *intrinsic evaluation*, that is, by means of measures that assess the quality of the similarity compared to human judgments (Section 7.1), and
- *in vivo* or *extrinsic evaluation*, that is, where the quality of a similarity approach is evaluated by measuring the impact on the performance of an application when integrating such approach therein (Section 7.2).

In vitro evaluation may suffer from several issues, such as the subjectivity of the annotation and the representativeness of the dataset. In contrast, in vivo evaluation is ideal, in that it shows a clear effectiveness on a separate application.

7.1 In vitro evaluation

We introduce the key measures used in the literature in Section 7.1.1 and overview several manually annotated datasets to which the measures are applied in Section 7.1.2.

7.1.1 Measures

The quality of a semantic similarity measure, be it knowledge-based or distributional, is estimated by computing a correlation coefficient between the similarity results obtained with the measure and those indicated by human annotators. Different measures can be employed for determining the correlation between variables. The two most common correlation measures are: Pearson's coefficient and Sperman's rho coefficient.

Pearson correlation coefficient. The Pearson correlation coefficient, also called Pearson product-momentum correlation coefficient, is a popular measure employed for computing the degree of correlation between two variables X and Y:

$$r = \frac{cov(X, Y)}{\sigma_X \sigma_Y} \tag{26}$$

where $cov(X, Y)$ is the covariance between X and Y, and σ_X is the standard deviation of X (analogously for Y). For a dataset of n word pairs for which the similarity has to be calculated, the following formula is computed:

$$r = \frac{\sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \overline{x})^2} \sqrt{\sum_{i=1}^{n} (y_i - \overline{y})^2}} \tag{27}$$

where the dataset is made up of n word pairs $\{(w_i, w'_i)\}_{i=1}^{n}$, x_i is the similarity between w_i and w'_i computed with the similarity measure under evaluation, y_i is the similarity provided by the human annotators for the same word pair and $\overline{x}$ is the mean of all values x_i.

Spearman's rank correlation. In determining the effectiveness of word similarity, the Pearson correlation coefficient has sometimes been criticized because it determines how well the similarity measure fits the values provided in the gold-standard, humanly produced datasets. However, it is suggested that for similarity it might be more important to determine how well the ranks of the similarity values correlate, which makes the measure non-parametric, that is, independent of the underlying distribution of the data. This can be calculated with Spearman's rank correlation, which is Pearson's coefficient applied to the ranked variables $rank_X$ and $rank_Y$ of X and Y. Given a dataset of n word pairs as above, the following formula can be computed:

$$\rho = 1 - \frac{6 \sum_{i=1}^{n} (rank(x_i) - rank(y_i))^2}{n(n^2 - 1)} \tag{28}$$

where $rank(x_i)$ is the rank value of the i-th item according to the similarity measure under evaluation and $rank(y_i)$ is the rank value of the same item according to the overall ranking of similarity scores provided by human annotators in the evaluation dataset.

7.1.2 Datasets

Several datasets have been created as evaluation benchmarks for semantic similarity. Here we overview the most popular of these datasets.

Rubenstein & Goodenough (RG-65) and its translations. A dataset made up of 65 pairs of nouns selected to cover several types of semantic similarities was created by Rubenstein and Goodenough (1965). Annotators were asked to assign each pair with a value between 0.0 and 4.0 where the higher the score, the higher the similarity. Due to the paucity of datasets in languages other than English, some of these datasets have been entirely or partially translated into various languages obtaining similar scores, including German (Gurevych 2005), French (Joubarne and Inkpen 2011), Spanish (Camacho-Collados, Pilehvar, and Navigli 2015), Portuguese (Granada, Trojahn, and Vieira 2014) and many other languages (Bengio *et al.* 2018).

Miller & Charles (MC-30). From these 65 word pairs, Miller and Charles (1991) selected a subset of 30 noun pairs, dividing it into three categories depending on the degree of similarity. Annotators were asked to evaluate the similarity of meaning, thus producing a new set of ratings.

WordSim-353. To make available a larger evaluation dataset, Finkelstein *et al.* (2002) elaborated a list of 353 pairs of nouns representing different degrees of similarity. To distinguish between similarity and relatedness pairs and set a level playing field in the evaluation of approaches which perform best in one of the two directions, WordSim-353 was divided into two sets (Agirre *et al.* 2009), one containing 203 similar pairs of nouns (WordSim-203) and one containing 252 related nouns (WordRel-252).[k]

BLESS. One of the first datasets specifically designed to intrinsically test the quality of distributional space models is BLESS. This dataset includes 200 distinct English nouns and, for each of these, several semantically related words (Baroni and Lenci 2011).

Yang and Powers (YP). Because traditional datasets work with noun pairs, Yang and Powers (2006) released a verb similarity dataset which includes a total of 144 pairs of verbs.

SimVerb-3500. To provide a larger and more comprehensive gold standard dataset for verb pairs, Gerz *et al.* (2016) produced a resource providing scores for 3500 verb pairs.

MEN. Bruni *et al.* (2014) released a dataset of 3,000 word pairs with semantic relatedness ratings in the range [0, 1], obtained by crowdsourcing using Amazon Mechanical Turk.

Rel-122. A further dataset for semantic similarity was proposed by Szumlanski (2013) who compiled a new set including 122 pairs of nouns.

SimLex-999. One of the largest resources providing word similarity scores was produced by Hill (2015). This dataset distinguishes clearly between similarity and relatedness by assigning related items with lower scores. Furthermore, it contains a large and differentiated set of adjectives, nouns and verbs, thus enabling a fine-grained evaluation of the performance.

Cross-lingual datasets. Camacho-Collados *et al.* (2015) addressed the issue of comparing words across languages by providing fifteen cross-lingual datasets which contain items for any pair of the English, French, German, Spanish, Portuguese and Farsi languages. More data aimed at multilingual and cross-lingual similarity were made available at SemEval-2017 (Camacho-Collados, Pilehvar, and Navigli 2017).

7.2 In vivo evaluation

An alternative way of evaluating and comparing semantic similarity measures is by integrating them into an end-to-end application and then measuring the performance change (hopefully, the improvement) of the latter compared to the baseline performance. Word and sense similarity are, indeed, intermediate tasks that lend themselves to the integration into an application. Among the most popular applications we cite:

(1) **Information retrieval:** word similarity has been applied historically to Information Retrieval (IR) since the development of the SMART system (Salton and Lesk 1968). More recent work performs IR using ESA (Gabrilovich and Markovitch 2007), or employs similarity in geographic IR (Janowicz, Raubal, and Kuhn 2011), in semantically enhanced IR (Hliaoutakis *et al.* 2006) and domain-oriented IR (Ye *et al.* 2016).

(2) **Text classification:** word similarity has also been used for classification since the early days (Rocchio 1971). More recently, word embeddings have been used to compute the similarity between words in the text classification task (Liu *et al.* 2018). Topical word, that is, context-based, representations (Liu *et al.* 2015) and bag-of-embeddings representations (Jin *et al.* 2016) have also been proposed which achieve performance improvement in text classification: NASARI embeddings have been used to create rich representations of documents and perform an improved classification of text (Sinoara *et al.* 2019).

(3) **Word sense disambiguation:** in order to choose the right sense of a given word, the similarity between sense vector representations, such as those available in NASARI, and the other words in the context has been computed (Camacho-Collados *et al.* 2016). Word similarity has been employed also in the context of word sense induction, that is, the task

[k] `http://alfonseca.org/eng/research/wordsim353.html`.

of automatically determining the senses of words (Schütze 1998), with the creation of the so-called multi-prototype embeddings (Tian *et al.* 2014; Liu *et al.* 2015; Pelevina *et al.* 2016).

(4) **Text summarization:** word similarity has been used to determine the correlation between summary pairs (Lin and Hovy 2003).
(5) **Machine translation:** word similarity has been proposed as a tool to improve the well-known issues with the n-gram overlap-based evaluation performed with the BLEU score (Banerjee and Lavie 2005; Castillo and Estrella 2012).
(6) **Synonym identification:** word and sense similarity have been used in the literature (Pilehvar *et al.* 2013) to determine the best synonym in the TOEFL synonymy recognition task (Landauer and Dumais 1997).
(7) **Coreference resolution:** word similarity has been used for anaphora resolution (Gasperin and Vieira 2004); in a more complex coreference resolution system, measures of semantic relatedness have been used as features for classifying referring expressions (Strube and Ponzetto 2006).
(8) **Malapropism detection:** semantic similarity has been employed to detect and correct malapropisms (Budanitsky and Hirst 2006), that is, real-word spelling errors.
(9) **Dictionary linking:** a key effort in electronic lexicography and related fields concerns linking dictionary entries. This can be done by determining the similarity between senses in different entries of two dictionaries (Pilehvar, Jurgens, and Navigli 2014). In like manner, sense clustering can be performed to reduce the dictionary granularity based on semantic similarity between concepts (Navigli 2006).

8. Conclusions and future directions

Similarity is at the core of NLP, in that all kinds of linguistic items need to be compared to perform tasks at all levels. In this article we have provided an overview of semantic similarity at the word and sense level. We have introduced the two main approaches to similarity, that is, the knowledge-based approach and the distributional approach, including recent developments based on neural networks, and we have described the various evaluation settings, including the most popular datasets in the field.

The reader might be wondering when and why sense representations should be preferred over word representations. There are several points to consider when facing this choice, which are the following:

(1) sense representations are preferable in any case if they improve the performance in word similarity tasks (cf. Section 7).
(2) sense representations provide linkage to existing lexical knowledge resources such as WordNet and BabelNet, in some cases performing disambiguation implicitly (Pilehvar *et al.* 2013; Camacho-Collados *et al.* 2016).
(3) meaning representations often enable multilingual or cross-lingual similarity without the need either to retrain the word embeddings or to use bilingual/multilingual lexical embeddings, which work well with predominant senses, but less so with infrequent meanings.

There are several promising directions in which word and sense similarity can develop. The first, and probably the most important, is *multilinguality*: being able to compare linguistic items in multiple languages is still understudied, even though recent developments in bilingual and multilingual embeddings are very promising (Ammar *et al.* 2016; Smith *et al.* 2017; Conneau *et al.* 2018). More interestingly, being able to contrast words and senses across languages, that is, by pairing items in different languages, might boost current new developments in fields such as machine translation (Lample *et al.* 2018; Artetxe, Labaka, and Agirre 2018).

A second important direction is the creation of *solid benchmarks for evaluating linguistic items* at different levels. Currently, there are several datasets of different types. Unfortunately, not all datasets contain the same kind of information. For instance, some are more geared towards measuring the degree of similarity between pairs, while still others more towards relatedness, others mix the two kinds of correlation in the same data. It is not obvious whether a solid in-vitro evaluation benchmark would be better than a single effective end-to-end application, probably because similarity measures tend to behave differently depending on the task and the nature of the data under study.

Importantly, we have seen that similarity is in many cases achieved with two ingredients: an effective representation of the linguistic item and a measure that exploits that representation to determine similarity. As the two main approaches to similarity tend to use different kinds of information, we believe strongly that a key further research direction should be to achieve a *tighter integration of knowledge with distributional semantics*, that is, to enhance neural architectures with explicit knowledge. This can be achieved, on the one hand, by leveraging multilingual lexical-semantic knowledge resources such as BabelNet, and, on the other hand, by learning relationships between distributional information in multiple languages, such as multilingual embeddings which share the same semantic vector space.

A final important direction, related to the above point, concerns the *adaptability* of the similarity approaches to different domains, tasks and applications. This direction is underexplored and calls for more investigation in order to establish the extent to which representations and measures have to be adapted to each new need (Bollegala, Maehara, and Kawarabayashi 2015; Prathusha, Liang, and Sethares 2018; Yang, Lu, and Zheng 2019).

9. Further reading

For additional information on the topic, we refer the reader to other surveys in word similarity (Mihalcea, Corley, and Strapparava 2006) and relatedness (Zhang, Gentile, and Ciravegna 2013), word and sense embeddings (Camacho-Collados and Pilehvar 2018) and semantic vector space models (Turney and Pantel 2010).

Acknowledgements. This work was funded by the ELEXIS project (contract no. 731015) under the European Union's Horizon 2020 research and innovation programme.

References

Agirre E., Alfonseca E., Hall K., Kravalova J., Paşca M. and Soroa A. (2009). 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*, pp. 19–27.

Ammar W., Mulcaire G., Tsvetkov Y., Lample G., Dyer C. and Smith N.A. (2016). *Massively multilingual word embeddings.* CoRR, abs/1602.01925.

Artetxe M., Labaka G. and Agirre E. (2018). Unsupervised statistical machine translation. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3632–3642.

Banerjee S. and Pedersen T. (2003). Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of IJCAI*, vol. 3, pp. 805–810.

Banerjee S. and Lavie A. (2005). METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72.

Baroni M. and Lenci A. (2011). How we BLESSed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics, Association for Computational Linguistics*, pp. 1–10.

Baroni M., Dinu G. and Kruszewski G. (2014). 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)*, pp. 238–247.

Barzegar, S., Davis, B., Zarrouk, M., Handschuh, S. and Freitas, A. (2018). SemR-11: a multi-lingual gold-standard for semantic similarity and relatedness for eleven languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

Bengio Y., Ducharme R., Vincent P. and Jauvin C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research* **3**(Feb), 1137–1155.

Bollegala D., Maehara T. and Kawarabayashi K.I. (2015). Unsupervised cross-domain word representation learning. In *Proceedings of the ACL 2015*, pp. 730–740.

Bruni E., Tran N.K. and Baroni M. (2014). Multimodal distributional semantics. *Journal of Artificial Intelligence Research* **49**, 1–47.

Budanitsky A. and Hirst G. (2001). Semantic distance in WordNet: an experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and Other Lexical Resources*.

Budanitsky A. and Hirst G. (2006). Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics* **32**(1), 13–47.

Camacho-Collados J., Pilehvar M.T. and Navigli R. (2015). A framework for the construction of monolingual and cross-lingual word similarity datasets. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers*, vol. 2, pp. 1–7.

Camacho-Collados J., Pilehvar M.T. and Navigli R. (2016). Nasari: integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence Journal* **240**, 36–64.

Camacho-Collados J., Pilehvar M.T. and Navigli R. (2017). SemEval-2017 Task 2: multilingual and cross-lingual semantic word similarity. In *Proceedings of SemEval@ACL 2017*, pp. 15–26.

Camacho-Collados J. and Pilehvar M.T. (2018). From word to sense embeddings: a survey on vector representations of meaning. *Journal of Artificial Intelligence Research* **63**, 743–788.

Castillo J. and Estrella P. (2012). Semantic textual similarity for MT evaluation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pp. 52–58.

Church K.W. and Hanks P. (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics* **16**(1), 22–29.

Collobert R., Weston J., Bottou L., Karlen M., Kavukcuoglu K. and Kuksa P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research* **12**(Aug), 2493–2537.

Conneau A., Lample G., Ranzato M.A., Denoyer L. and Jégou H. (2017). Word translation without parallel data. In *Proceedings of ICLR 2018*.

Deerwester S., Dumais S.T., Furnas G.W., Landauer T.K. and Harshman R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science* **41**(6), 391–407.

Devlin J., Chang M.W., Kenton Lee K. and Toutanova K. (2018). Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Fellbaum C. (1998). *WordNet: An Electronic Lexical Database*. Cambridge, MA, USA: MIT Press.

Ferraresi A., Zanchetta E., Marco Baroni M. and Bernardini S. (2008). Introducing and evaluating ukWaC, a very large web-derived corpus of English. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4)*, pp. 47–54.

Finkelstein L., Gabrilovich E., Matias Y., Rivlin E., Solan Z. and Wolfman G., Ruppin E. (2002). Placing search in context: the concept revisited. *ACM Transactions on Information Systems* **20**(1), 116–131.

Firth J.R. (1957). A synopsis of linguistic theory. In *Studies in Linguistic Analysis*, Blackwell, Oxford, pp. 1930–1955.

Gabrilovich E. and Markovitch S. (2007). Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proceedings of IJCAI*, vol. 7, pp. 1606–1611.

Gasperin C. and Vieira R. (2004). Using word similarity lists for resolving indirect anaphora. In *Proceedings of the Conference on Reference Resolution and Its Applications*.

Gerz D., Vulić I., Hill F., Reichart R. and Korhonen A. (2016). Simverb-3500: a large-scale evaluation set of verb similarity. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2173–2182.

Graff D., Kong J., Chen K. and Maeda K. (2003). *English Gigaword*. Philadelphia: Linguistic Data Consortium.

Granada R., Trojahn C. and Vieira R. (2014). Comparing semantic relatedness between word pairs in Portuguese using Wikipedia. In *International Conference on Computational Processing of the Portuguese Language*, pp. 170–175.

Grefenstette G. (1994). *Explorations in Automatic Thesaurus Discovery*. Norwell, MA: Kluwer Academic Publishers.

Gurevych I. (2005). Using the structure of a conceptual network in computing semantic relatedness. In *Proceedings of International Conference on Natural Language Processing*, pp. 767–778.

Harris Z.S. (1954). Distributional structure. *Word* **10**(2–3), 146–162.

Han L. and Finin T. (2013). *UMBC webBase corpus*.

Haveliwala. (2002). Topic-sensitive PageRank. In *Proceedings of the 11th International Conference on World Wide Web*, pp. 517–526.

Hassan S.H. and Mihalcea R. (2011). Semantic relatedness using salient semantic analysis. In *Proceedings of Twenty-Fifth AAAI Conference on Artificial Intelligence*.

Hill F., Reichart R. and Korhonen A. (2015). SimLex-999: evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics* **41**(4), 665–695.

Hliaoutakis, A., Varelas, G., Voutsakis, E., Petrakis, E.G. and Milios, E. (2006). Information retrieval by semantic similarity. *International Journal on Semantic Web and Information Systems (IJSWIS)* **2**(3), 55–73.

Hovy E., Navigli R. and Ponzetto S.P. (2013). Collaboratively built semi-structured content and Artificial Intelligence: the story so far. *Artificial Intelligence Journal* **194**, 2–27.

Hughes T. and Ramage D. (2007). 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)*, pp. 581–589.

Iacobacci I., Pilehvar M.T. and Navigli R. (2015). SensEmbed: learning sense embeddings for word and relational similarity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, pp. 95–105.

Janowicz K., Raubal M. and Kuhn W. (2011). The semantics of similarity in geographic information retrieval. *Journal of Spatial Information Science* **2011**(2), 29–57.

Jarmasz M. and Szpakowicz S. (2003). Roget's thesaurus and semantic similarity. In *Proceedings Conference on Recent Advances in Natural Language Processing*, pp. 212–219.

Jiang J.J. and Conrath D.W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the 10th International Conference on Research in Computational Linguistics (ROCLING X)*.

Jin P., Zhang Y., Chen X. and Xia Y. 2016. *Bag-of-embeddings for text classification.* In *Proceedings of IJCAI*, vol. 16, pp. 2824-2830.

Joubarne C. and Inkpen D. (2011). Comparison of semantic similarity for different languages using the Google N-gram corpus and second-order co-occurrence measures. In *Advances in Artificial Intelligence*, pp. 216–221.

Joulin A., Grave E., Bojanowski P. and Mikolov T. (2017). Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2*, pp. 427–431.

Jurafsky D. (2000). *Speech & Language Processing*. India: Pearson Education.

Jurgens D. (2016). Cross level semantic similarity: an evaluation framework for universal measures of similarity. *Language Resources and Evaluation* **50**(1), 5–33.

Lafon P. (1980) Sur la variabilité de la fréquence des formes dans un corpus *Mots* **1**(1), 127–165.

Lample G., Ott M., Conneau A., Denoyer L. and Ranzato M.A. (2018). Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 5039–5049.

Landauer T.K. and Dumais S.T. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review* **104**(2), 211–240.

Leacock C., Miller G.A. and Chodorow M. (1998). Using corpus statistics and WordNet relations for sense identification. *Computational Linguistics* **24**(1), 147–165.

Levy O. and Goldberg Y. (2014). Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, pp. 2177–2185.

Lesk M. (1986). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of SIGDOC*, pp. 24–26.

Lin D. (1998). An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning (ICML)*, pp. 296–304.

Lin C.Y. and Hovy E. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

Liu Y., Liu Z., Chua T.S. and Sun M. (2015). Topical word embeddings. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Liu Q., Huang H., Gao Y., Wei X., Tian Y. and Liu L. (2018). Task-oriented word embedding for text classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 2023–2032.

Mihalcea R., Corley C. and Strapparava C. (2006). Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of AAAI*, vol. 6, pp. 775–780.

Miller G.A., Beckwith R., Fellbaum C., Gross D. and Miller K.J. (1990). Introduction to WordNet: an on-line lexical database. *International Journal of Lexicography* **3**(4), 235–244.

Miller G.A. and Charles W.G. (1991). Contextual correlates of semantic similarity. Language and Cognitive Processes **6**(1), 1–28.

Mikolov T., Chen K., Corrado G. and Dean J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Milne D.N. and Witten I.H. (2008). An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy*.

Moro A. and Navigli R. (2013). Integrating syntactic and semantic analysis into the open information extraction paradigm. In *Proceedings of IJCAI*, pp. 2148–2154.

Moro M., Raganato A. and Navigli R. (2014). Entity linking meets word sense disambiguation: a unified approach. *TACL* **2**, 231–244.

Moro A. and Navigli R. (2015). SemEval-2015 Task 13: multilingual all-words sense disambiguation and entity linking. In *Proceedings of SemEval*, pp. 288–297.

Moussallem D., Wauer M. and Ngonga Ngomo A.C. (2018). Machine translation using semantic web technologies: a survey. *Journal of Web Semantics* **51**, 1–19.

Navigli R. (2006). Meaningful clustering of senses helps boost word sense disambiguation performance. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pp. 105–112.

Navigli R. and Ponzetto S.P. (2012). BabelNet: the automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence Journal* **193**, 217–250.

Navigli R., Faralli S., Soroa A., de Lacalle O. and Agirre E. 2011. Two birds with one stone: learning semantic models for text categorization and word sense disambiguation. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pp. 2317–2320.

Navigli R., Jurgens D. and Vannella D. (2013). SemEval-2013 Task 12: multilingual word sense disambiguation. In *Proceedings of SemEval-2013*, vol. 2, pp. 222–231.

Page L., Brin S., Motwani R. and Winograd T. (1998). The PageRank citation ranking: bringing order to the web. In *Proceedings of the 7th International World Wide Web Conference, Brisbane, Australia*, pp. 161–172.

Papineni K., Roukos S., Ward T. and Zhu W.J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 311–318.

Pelevina, M., Arefyev, N., Biemann, C. and Panchenko, A. (2016). Making sense of word embeddings. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, p. 174–183.

Pennington, J., Socher, R. and Manning, C.(2014). Glove: global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543.

Peters M.E., Neumann M., Iyyer M., Gardner M., Clark C., Lee K. and Zettlemoyer L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237.

Pilehvar M.T., and Camacho-Collados J. (2019). WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1267–1273.

Pilehvar M.T., Jurgens D. and Navigli R. (2013). Align, disambiguate and walk: a unified approach for measuring semantic similarity. In *Proceedings of the Association of Computational Linguistics*, pp. 1341–1351.

Pilehvar M.T, Jurgens D. and Navigli R. (2014). A robust approach to aligning heterogeneous lexical resources. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, pp. 468–478.

Pilehvar M.T. and Collier N. (2016). De-conflated semantic representations. In *Proceedings of EMNLP 2016*, pp. 1680–1690.

Prathusha K.S., Liang Y. and Sethares W.A. (2018). Domain adapted word embeddings for improved sentiment classification, In *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP*, pp. 51–59.

Rada R., Mili H., Bicknell E. and Blettner M. (1989). Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics* **19**(1), 17–30.

Resnik P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pp. 448–453.

Rocchio J.J. (1971). The smart retrieval system: experiments in automatic document processing. In *Relevance Feedback in Information Retrieval*, Prentice Hall, Upper Saddle River, pp. 313–323.

Rothe S. and Schütze H. (1995). Autoextend: extending word embeddings to embeddings for synsets and lexemes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers),* vol. 1, pp. 1793–1803.

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

Rumelhart D.E., Hinton G.E. and Williams R.J. (1988). Learning representations by back-propagating errors. *Cognitive Modeling* **5**(3).

Salton G. and Lesk M.E. (1968). Computer evaluation of indexing and text processing. *Journal of the ACM (JACM)* **15**(1), 8–36.

Schütze H. (1998). Automatic word sense discrimination. *Computational Linguistics* **24**(1), 97–123.

Smith S. L., Turban D.H.P, Hamblin S. and Hammerla N.Y. (2017). Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *5th International Conference on Learning Representations (ICLR)*.

Sinoara R.A., Camacho-Collados J., Rossi R.G., Navigli R., and Rezende S.O. (2019). Knowledge-enhanced document embeddings for text classification. *Knowledge-Based Systems*, 163, 955–971.

Strube M. and Ponzetto S.P. (2006). Wikirelate! computing semantic relatedness using Wikipedia, In *Proceedings of the Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference*, July 16–20, 2006, Boston, Massachusetts, USA, pp. 1419–1424.

Szumlanski S., Gomez F. and Sims V.K. (2013). A new set of norms for semantic relatedness measures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, pp. 890–895.

Tian, F., Dai, H., Bian, J., Gao, B., Zhang, R., Chen, E. and Liu, T.Y. (2013). A probabilistic model for learning multi-prototype word embeddings. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 151–160.

Turney P. and Pantel P. (2010). From frequency to meaning: vector space models of semantics. *Journal of Artificial Intelligence Research* **37**(1), 141–188.

Wu Z. and Palmer M. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, pp. 133–138.

Yang, W., Lu, W. and Zheng, V.W. (2019). A simple regularization-based algorithm for learning cross-domain word embeddings. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2898–2904.

Yang D. and Powers D. M. (2006). Verb similarity on the taxonomy of WordNet. Masaryk University.

Ye, X., Shen, H., Ma, X., Bunescu, R. and Liu, C. (2016). From word embeddings to document similarities for improved information retrieval in software engineering. In *Proceedings of the 38th International Conference on Software Engineering*, pp. 404–415.

Zesch T., Müller C. and Gurevych I. (2008). Using wiktionary for computing semantic relatedness. In *Proceedings of AAAI*, vol. 8, pp. 861–866.

Zhang Z., Gentile A.L. and Ciravegna F. (2013). Recent advances in methods of lexical semantic relatedness–a survey. *Natural Language Engineering* **19**(4), 411–479.

Cite this article: Navigli R and Martelli F (2019). An overview of word and sense similarity. *Natural Language Engineering* **25**, 693–714. https://doi.org/10.1017/S1351324919000305