Regular Paper

DBSJ Journal
Vol. 14, Article No. 1, March 2016

Measuring Semantic Similarity and Relatedness with Distributional and Knowledge-based Approaches

Christoph LOFI

This paper provides a survey of different techniques for measuring semantic similarity and relatedness of word pairs. This covers both knowledge-based approaches exploiting taxonomies like WordNet, and corpus-based approaches which rely on distributional statistics. We introduce these techniques, provide evaluations of their result performance, and discuss their merits and shortcomings. A special focus is on word embeddings, a new technique which recently became popular with the AI community. While word embeddings are not fully understood yet, they show promising results for similarity tasks, and may also be suitable for capturing significantly more complex features like relational similarity.

1. Introduction

Semantic similarity plays a central role in how humans process knowledge, and serves as an organization principle for classifying objects, formulating concepts, and performing generalizations and abstractions [1]. Therefore, it is not surprising that semantic similarity also plays a key-role in many information management tasks, ranging from similarity navigation in E-Commerce [2], community mining [3], word sense disambiguation [4], cross-document co-reference [5], and many more. Being able to effectively measure similarity is therefore a central challenge when dealing with the flood of unstructured text documents contained in BigData repositories. Simplified, semantic similarity classifies concepts into different types or kinds, and similarity measures quantify how alike two concepts are. From an ontological point of view, semantic similarity quantifies the perceived distance between two concepts with respect to their type (e.g., "horse" is very similar to "donkey" because they are both equine animals) or function (i.e. "horse" is somewhat similar to "camel" as both can be used for riding). In contrast, relatedness measurements quantify the relatedness of objects with respect to other relationship types, e.g. "harness" is highly related to "horse" but is not similar (harnesses and horses don't share a common type nor function, but have other strong relationships. i.e. a harness is often used on a horses). Both concepts are neither exclusive nor independent, and are often not cleanly differentiated by neither people nor algorithms.

The challenge of implementing meaningful measures for similarity and relatedness is spiked with several hurdles: ontologies currently available usually do not show the required degree of completeness with respect to the contained concepts and relationships. Also, ontologies lack a meaningful sense of "perceived distance", e.g. it is unclear how much focus is lost when moving along a relationship edge. For this reason, corpus-based approaches have emerged which rely on exploiting word statistics in large natural text corpora for approximating perceived similarity and relatedness without using explicit ontological knowledge. While these approaches can easily be adapted to any domain where a sufficiently large corpus is available, they usually lack the semantic expressiveness to differentiate between relatedness and similarity and just return a combined measure.

In this paper, we will provide an in-depth overview of existing similarity and relatedness measures, and discuss their strength and weaknesses. Finally, we benchmark them on several established test corpora, and discuss their strength and weaknesses.

2. Computing Similarity and Relatedness

In this section, we will present the current state-of-the-art for assessing similarity and relatedness between a pair of words. Such approaches can roughly be classified in *knowledge-based techniques*, which rely on a given ontology or taxonomy, and corpus-based approaches which use a large corpus of natural language text (and of course, there are also hybrids between both). Corpus-based techniques be further classified into *simple distributional approaches* exploiting co-occurrences of words, and approaches based on *dense vector representations* which are usually the result of applying dimensionally reduction techniques to vector-based distributional approaches.

In the following, we will outline the most relevant techniques for each type, and then we will present the respective performance evaluation in chapter 3.

2.1 Knowledge-based Approaches

The approaches presented in this section exploit an existing ontology or taxonomy. Such ontologies are usually manually or semi-manually created and maintained, and can be very costly. Therefore, only few domains will have a suitable ontology which limits the applicability of similarity measures based on one. There are some mostly manually created general domain ontologies like WordNet [6] or Freebase [7], but also some which rely on automatic information extraction (from manually created sources) like DBpedia [8] or Yago [9]. Ontologies represent each concept as a node, which is linked by edges to other concepts representing relationships between them. Knowledge-based similarity measures usually only exploit taxonomic information, i.e. the hierarchical classification of all concepts via "is-a" relationships. Here, the basic intuition is that two concepts are more similar if they are closer to each other considering taxonomic relationships. This leads to simple similarity measures which either count the number of nodes or the number of edges (shortest path approaches) separating words in a taxonomy. Such techniques will usually yield low result quality, as they ignore the inhomogeneous granularity of taxonomical relationships (e.g., the semantic distance from "animal" to "living thing" is significantly larger than the distance from "poodle" to "dog"). Approaches like Leacock & Chodorow [10] try to rectify this by setting the path length in relation to the maximum depth of the taxonomy, and is given by:

$$sim_{lch}(w_1, w_2) = -log \frac{min\_lenght(w_1, w_2)}{2 * Depth_{max}}$$

In a similar fashion, the approach of Wu & Palmer [11] measures the depth of two concepts in a taxonomy in relation to the least common subsumer node (LCS):

$$sim_{wup}(w_1, w_2) = 2 * \frac{depth\big(LCS(w_1, w_2)\big)}{depth(w_1) + depth(w_2)}$$

Such approaches purely consider the structure of a taxonomy, and thus still have trouble judging the perceived semantic distance of a relationship. Therefore, several approaches which also consider the information content (IC) have been developed. The information content of a concept (which is represented by the word w) is given by the probability of encountering the concept in an large natural language text corpus (i.e. frequent words are less informative than infrequent words), and is given by $IC(w) = -\log P(c)$. This probability can be approximated by counting word occurrences in the corpus. The approach of Resnik [12] was one of the first to adapt this idea, relying on the information content of the least common subsumer (LCS) node. Therefore, this technique can be considered a hybrid approach between corpus- and knowledge-based approaches, and is given by:

$$sim_{res}(w_1, w_2) = -log\ P\big(LCS(w_1, w_2)\big) = IC\big(LCS(w_1, w_2)\big)$$

Later works by Lin [13] and Jiang & Conrad [14] expand on this idea, and slightly improve scaling and normalization of the similarity measures. They are given by:

$$sim_{lin}(w_1, w_2) = 2 * \frac{IC\big(LCS(w_1, w_2)\big)}{IC(w_1) + IC(w_2)}$$

$$sim_{jc}(w_1, w_2) = \frac{1}{IC(w_1) + IC(w_2) - 2 * IC\big(LCS(w_1, w_2)\big)}$$

The measures presented above are only effective for measuring similarity, and fail to also capture relatedness (this can be seen as an advantage or disadvantage depending on the use case scenario as most approaches which support relatedness cannot differentiate it from similarity. However, for many applications, it is beneficial to have both similarity and relatedness measures available). This can be explained by the limited semantics of the information content measure, and is considered by alternative definitions like the intrinsic IC [15], which is given by the number of subconcepts and the total number of concepts in a taxonomy (max_C):

$$iIC(w) = 1 - \frac{log(\#subconcepts(w) + 1)}{log(max_{con})}$$

This means that a concept has a maximal intrinsic IC if it cannot be further differentiated, i.e. is a leaf node in the taxonomy. This can be used as basis for the extended IC [16], which also tries to quantify information content with respect to relatedness. Here, the base idea is that many structured knowledge bases are not just simply taxonomies containing only subsumption relationships, but also have additional relationship types (like *is-part-of*, *is-used-in*, *is-located-in*, etc.). Extended IC considers all m relationship types connecting a given concept to any other one, and computes the average iIC for all n concepts at the other end of the relationships of a given type (i.e., for each $w_k \in W_{R_J}$. In the following formula, m is the number of different relationship types the concept w has to other concepts, and W_{R_J} is the set of concepts related to w via a given relationship type R_J. Basically, EIC represents the average information content considering all relationship types):

$$EIC(w) = \sum_{j=1}^{m} \frac{\sum_{k=1}^{n} iIC\big(w_k \in W_{R_J}\big)}{\big|W_{R_J}\big|}$$

In a least step, extended IC is combined and weighted with the intrinsic IC, i.e.:

$$eIC(w) = \alpha * iIC(w) + \beta * EIC(w)$$

This leads to the FaITH [16] measure, which is similar to the Jiang and Conrad definition, but uses extended IC instead:

$$sim_{FaITH}(w_1, w_2) = \frac{1}{eIC(w_1) + eIC(w_2) - 2 * eIC\big(LCS(w_1, w_2)\big)}$$

While this definition solves many weaknesses (i.e. considers features of concepts given by relationships as proposed by Tversky [1]), the benchmarks in later chapters will show that this approach still has some shortcomings: even though the theory is sound, the approach is limited by the incompleteness of the used ontology. For example, the WordNet ontology is quite good from a taxonomical point of view (e.g., it contains "poodle *is_a* dog"), but contains only a small selection of non-taxonomic relationships (e.g., it does not contain "harness *is_used_for_riding* horses"). This observation leads to purely corpus-based approaches, which try to circumvent the need for an ontology or taxonomy altogether.

2.2 Simple Distributional Approaches

As there is a lack of high quality ontologies in many domains, corpus-based approaches try to measure similarity and relatedness by exploiting statistics over large text corpora. This is especially interesting in domains which change and adapt quickly, and where frequently new concepts and words are introduced (as for example in the e-commerce domain.) A central assumption of corpus-based techniques is the distributional hypothesis [17], which claims that words that occur in similar contexts in a large text corpus also have similar or related semantics. A context is often given by just a small chunk of text (e.g., several consecutive words or a single sentence). Resulting from the distributional hypothesis, current approaches have in common that they cannot distinguish similarity from relatedness, and treat both as being interchangeable.

A naïve implementation of the distributional hypothesis is using the PMI measure, i.e. Pairwise Mutual Information [18], which measures the ratio of the probability of pairwise occurrences of two words in the same context in relation to the probability of isolated occurrence of each

word. In many early works, as for example [19], a Web search engine based IR approach was used to implement PMI, and the context probabilities have been approximated by search term hit counts using search engines like Google or Yahoo. This allows for obtaining the required context hit statistics without the need to obtain and process a large text corpus locally. Several approaches for defining "context" are possible, as for example assuming each Web page is a context or assuming a context is a small sequence of words, or a search engine summary snippet. In [18], it is recommended to use the "near" keyword which used to be offered by some Web search engines for identifying pages where both words appear close in close proximity. The basic definition of PMI is (with p being the probability of a word w occurring in a context):

$$PMI(w_1, w_2) = log_2 \frac{p(w_1 \& w_2)}{p(w_1) * p(w_2)}$$

In [20], several such relatedness measures based on page counts are compared, like PMI, Jaccard coefficents, Simpson coefficient, and Dice coefficient using lexio-syntactic patters as model features. In [21], the result preview snippets of queries have been used to build a vector representation based on TF-IDF measures, and semantic similarity is measured using a Kernel function. While many of the approaches mentioned above showed promising results at their time, they have become less attractive in recent years as major Web search engine providers have limited the usage of their APIs (as discussed in [22]). For example, few current search engines still supports the "near" keyword. Also, many search engines limit the number of maximum retrievable results to a rather low number (e.g., 1000 for Google), thus hindering approaches which rely on summary snippets. Furthermore, hit counts in current search engines are significantly less reliable than a couple of years ago (because approximation techniques became popular). This also contributes to why approaches relying on Web hit counts perform so poorly in the benchmarks presented in later chapters, while the same techniques used on a local (smaller) corpus perform significantly better. Unfortunately, further research on web-based measures to further quantify the effects of (proprietary) hit count approximations and the limitations to snippet results are no longer possible as these variables cannot be controlled in more in modern search engines. Therefore, current implementations of search-based techniques should rely on local search engines like Apache Lucene working on a corpus crawled from the Web or obtained in some other way. However, this dramatically increases the costs and complexity associated with these approaches.

Another straight-forward idea for corpus-based similarity and relatedness measurements is to present each word as a vector in high-dimensional space by exploiting again the distributional hypothesis. Then, the semantic similarity/relatedness of two words can be approximated by measuring the distance between two word vectors, e.g., by relying on the cosine or Cartesian vector distance.

In the bag-of-word (BOW) approach, each word vector for a word w is constructed by collecting all words appearing in a small (context) window centered around each occurrence of w, where the values represent the frequency of these occurrences. Therefore, the dimensionality of the word vectors match the number of different words in the corpus. The second approach is based on context windows (CW). Here, each context in which a given word w appears in is collected and considered being one dimension in the word vector (a context is a small window of words around a text occurrence with the actual word removed, e.g. "*the <term> barks*" could be a context for the word "dog"). The value of that dimension represents the frequency of the term in that context. This leads to vectors with very high dimensionality, but also high sparsity, as there is a very large number of possible contexts in a given corpus.

In [23], several such approaches for measuring similarity and relatedness are compared using a very large crawled Web corpus (~$1.6E^{12}$ words).

2.3 Dense Vector Representations

As mentioned in the previous section, a straight-forward technique for exploiting the distributional hypothesis is representing words as high dimensional sparse word count vectors. Even as such vectors can be used to measure semantic similarity and relatedness, they are unwieldy and complicate further semantic analysis of the corpus (like finding the dominant concepts of a domain). In many other domains, reducing the dimensionality of such vectors has been shown to have very beneficial results, and often increases the semantic usefulness of the vector representation.

Transferring this insight to word similarity vectors, a popular technique to achieve such a dimensionality reduction is to exploit matrix factorization for creating dense vector representations. The core idea here is to represent the corpus as a large sparse matrix M, where each row is associated with a word or concept, while the columns stand for a distributional measurement (as discussed in the last section). Common approaches for defining these features are for example the already mentioned bag-of-words counts, context window counts, or PMI values of words and context windows. Now, the matrix M is decomposed into at two matrixes with significantly reduced dimensionality such that their product approximates M as closely as possible. The matrix of interest has a row for each relevant concept (or word), and a smaller number of columns for the dominant latent features, while the other matrix shows the same reduced features and the original features of M (and is usually discarded). This result can be achieved by approaches like principal component analysis (PCA), or latent semantic analysis (LSA). A drawback of all dense vector representations (including neural embeddings presented later) is that they are computationally very expensive, often requiring multiple days for generating the vector representation. Considering their performance and versatility which goes beyond simple similarity measures, this is often a fair trade.

As an example for these approaches for computing word similarity, in [24], LSA [25] is applied to a corpus consisting of a Wikipedia dump and also the TASA corpus. Latent Semantic Analysis (LSA) is "a theory and method for extracting and representing the contextual-usage meaning of words by statistical computations applied to a large corpus of text". This is achieved by singular value decomposition

which results the eigenvectors and eigenvalues of the original word statistics matrix M, and then retaining only the n strongest vectors which will form the basis of the new reduced latent dimensions. Usually, 300 to 500 dimensions are retained, resulting in a dense word-feature matrix with reduced column dimensionality. Despite the reduced dimensionally, most semantics remain intact. In fact, the quality of such reduced representations is often higher as the reduction removes noise from the matrix, and generalizes concepts which is beneficial to many semantic tasks like automatic content analysis in information retrieval.

In the last few years there has been a surge of approaches proposing to build dense word vectors not by using matrix factorizing, but by using neural language models which have the training of a neural network at their core. Early neural language models were designed to predict the next word given a sequence of initial words of a sentence [26], [27] (as for example used in text input autocompletion) or to predict a nearby word given a cue word [28], [29]. Also, language models can be designed to automatically predict a translation into another language for a given text fragment [30], [31].

While neural language models can be designed for different tasks and trained with a variety of techniques, they share the trait that at their core, there is a dense vector representation of words which can be exploited for computing similarity. This representation is often referred to as "neural word embedding". The usefulness of these embedding may vary with respect to similarity computation based on the chosen technique and corpus.

The common process of training a neural language model is to learn real-valued embeddings for words in a predefined vocabulary. In each training step, a score for the current training example is computed based on the embeddings in their current state. This score is compared to the model's objective function, and then the error is propagated back to update both the model and the weights. At the end of this process, the embeddings should encode information that enables the model to optimally satisfy its objective [32].

Early neural language models like [26], [27] were trained using ordered sequences n of words extracted from a real-life text corpus, and the model was optimized to predict the n-th word given the first n-1 words. The model first represents the input as an ordered sequence of embeddings, which it then transforms into a single fixed length hidden representation in a neural network. Based on this representation, a probability distribution is computed over the vocabulary from which the model can sample and guess the next word. The model weights and embeddings are updated to maximize the probability of correct guesses for all sequences in the training corpus [33].

Later approaches adopted a significantly simpler approach for building a high quality word embedding which can be learned using a neural network without non-linear hidden layer (like the popular SGNS skip-gram negative sampling approach [28], [29]). These models are trained using 'windows' extracted from a natural language corpus (i.e. an unordered set of words which occur nearby in a text sequence in the corpus). The model is trained to predict, given a single word from the vocabulary, those words which will likely occur nearby it (i.e. share the same windows).

A current trend is to use neural language models for machine translation tasks, resulting in neural machine translation models. Their objective is to generate an appropriate sentence in a target language given the sentence in the source language (e.g., [30][30]). They are usually trained on a bi-lingual corpus of manually translated texts. As a by-product, such neural translation models also create word embeddings for both languages. One such approach is RNNenc [30], which uses a recurrent neural network to learn the model.

Measuring similarity and relatedness between words using embeddings is straight forward: after the embedding is created, the similarity measure is a result of the distance of the vectors of both words. However, beyond simple similarity and relatedness measures as discussed in this paper, many word embeddings show some very interesting and surprising properties: it seems that not only the cosine distance between vectors represents a measure for similarity and relatedness, but that also the difference vectors between a word pair carries meaningful semantics [34]. For example, the difference between the vector for "man" and "king" seems to represent the concept of being a ruler, and the vector of "woman" plus this concept vector will result in "queen". While the full extend and reasons for this behavior is not fully understood yet, these semantics have been impressively demonstrated for several examples like countries and their capitals, countries and their stereotypical food, or several grammatical relationships. Therefore, these neural word embeddings do not only encode similarity between words, but also relational similarity which can be used for analogical reasoning. Furthermore, the semantics of word embeddings created from corpora in different languages can be aligned: in [35], the authors show how by using only a few training examples for alignment, a surprisingly accurate dictionary between Chinese and English can be created automatically. A similar technique can also be used for neural word embeddings learned from a text corpus and a neural embedding learned from visual image data: in [36] the authors demonstrate that, to a certain extent, pictures can be automatically labeled by aligning two such different neural embeddings.

One of the core problems in neural embedding research is that neural networks tend to be opaque. While they may "just work" in many cases, it is often hard to explain why and how. Therefore, in [37], neural word embeddings are further examined from a theoretical point of view, and the authors were able to show that the process performed by the neural model implicitly resembles a matrix factorization not unlike the approaches discussed above. The matrix in question seems to be based on weighted and log-shifted PMI (SPMI) values of words and their surrounding word contexts. However, the neural network emphasizes certain weights during factorization in an attempt to perform generalizations, and this behavior seems to be the reason why the model allows for vector distance semantics while similar models which are directly based on matrix factorizations like SVD or LSA do not exhibit this property. In any case, neural word embeddings are a very promising technique which will require thorough future research. In

this paper, we will evaluate them in comparison with older and more established techniques for the simpler similarity/relatedness task, and the experiments will show that they show competitive performance and may serve as a replacement for older techniques.

3 Benchmarking and Experiments

3.1 Benchmarking Datasets

One of the core problems when evaluating algorithms computing semantic similarity or relatedness is that there is no undisputable correct result for a given measurement. Ultimately, similarity and relatedness has to be judged by human consensus. Therefore, semantic similarity may change across contexts, different cultural backgrounds, or subjective perceptions, or simplify over time. As a result, we rely on several standard gold datasets established in similarity research, and which have been elicited in controlled psychological studies. The first dataset (called RG65) was established by Rubenstein and Goodnough [38], and contains 65 word pairs. Each of these pairs have been judged for their semantic similarity only by 51 human subjects. The subjects had been instructed to ignore any other possible semantic relationships. In several additional studies, it has been shown that aggregated human judgments for this dataset usually show a very high degree of correlation with each other [12], [39], therefore despite its smalls size, this dataset is considered representative for evaluating semantic similarity. One of these studies resulted in the slightly smaller MC30 [39] gold dataset, covering 30 word pairs.

In addition, we use the larger WS353 dataset [40], containing 353 word pairs which each have been judged by 13 to 16 human subjects. However, in this datasets, no distinction between similarity and relatedness was made. Hence, a high aggregated value in the reference data may either represent a high degree of perceived similarity or a high degree of perceived relatedness (similar to the measurements of corpus-based approaches). In a similar fashion, the newer MEN test collection [41] was created, which contains 3,000 word pairs judged using crowdsourcing via Amazon Mechanical Turk.

For a long time, there was no established and controlled dataset which distinguishes between relatedness and similarity by design. Therefore, the WS353 dataset was manually split (in [23]) into three subsets: those pairs which are similar, those pairs which are related, and those which are neither related nor similar. From these, two new datasets where created: WS353-S containing 203 word pairs which are either similar or unrelated and dissimilar, and WS353-R containing 252 word pairs which are either related or unrelated and dissimilar.

This flaw of not distinguishing between similarity and relatedness is rectified with the newer Simlex-999 [42] reference. It contains 999 word pairs which can be only considered similar but not related, and was created with the help of 500 subjects hired via Amazon Mechanical Turk. Furthermore, the corpus pays respect to the theory that human similarity judgement differ with part-of-speech [43] and conceptual concreteness [42], resulting in 666 noun pairs, 222 verb pairs, and 111 adjective pairs. Furthermore, the pairs have a controlled concreteness distribution (e.g., [tree, bush] is more concrete, while [belief, faith] is less concrete). Each word pair is rated by 50 subjects in average. As the corpus is rather new, there have only limited experiences been reported yet. Nonetheless, this corpus is larger and more thoroughly designed than older corpora, and should be the preferred choice for current research on pure similarity measures.

3.2 Benchmark Results

In the following, we discuss benchmarking results for eight knowledge-based techniques, and fifteen corpus-based approaches. In several cases, we performed the benchmarks ourselves, but some techniques are very hard to replicate and/or have been tested on a specific corpora which is not open to the public. Therefore, we cannot provide benchmark results for all cells in in Table 1, and for some, we directly used results reported in literature. Cited original works often evaluated their approach on a single specific corpus, while others approaches have been evaluated on a different corpus. Therefore, the reported measurements represent the effects of both the technique used and the corpus used. This hampers direct comparison of approaches which were tested on different corpora. Still, some trends can be observed from these results.

The reported result values in Table 1 are the Pearson correlation between the judged similarity of humans (with respect to different benchmarking sets as discussed in the last subsection) and the measured similarity by a given technique. While there have been discussions if Spearman correlation might be a better representative for result quality, we found that both correlations show a comparable picture, and thus we report only Pearson correlation coefficients which is used more frequently in literature. Table 1 covers results for different test sets, and if a measurement was taken from literature, we refer to the source. Knowledge-based techniques have been implemented using the NLTK toolkit [44] with expectation of FaITH, which has been implemented using the tools provided by [16]. For word embeddings, we used the SGNS-based word2vec toolkit provided by [45] and the original implementation of GLOVE [29]. We used the default parameters suggested for each toolkit unless specified otherwise.

Knowledge-based approaches: Our observations can be summarized as follows: Knowledge-based approaches all achieve good results for the MC30 and RG65 datasets, with IC and eIC based approaches showing better performance than techniques only relying on taxonomical information (e.g., Leacock-Chodorow). However, when inspecting the larger test sets, this picture changes: by design, all knowledge-based methods but FaITH are not designed to measure relatedness, and therefore have a very low correlation with WS353-R (which only contains related and unrelated / dissimilar test pairs). But even FaiTH shows only low correlation values of around 0.38. This also affects the results of the full WS353 set which therefore is not representative for judging the algorithms' performance (same holds true for the MEN set). However, even when just considering WS353-S, which only contains similar and dissim-

ilar test pairs and therefore should work well, all approaches show subpar performance (with the simple Resnik measure being best among mediocre results). This can be explained by the limited nature of the RG65 test set, which mostly contains popular words which are covered in great detail in WordNet. In contrast, WS353-S also has some more obscure terms, which are not represented well in WordNet. In [41], it is suggested that the upper bound for correlation measures on WS353 is around 0.84, representing human correlation (for RG65, human correlation is around 0.98). Similar behavior can be observed on the SIMLEX datasets. This suggests that knowledge-based similarity measurement will fare even worse in specialized domains (like e-commerce) due to the limited nature of available taxonomies and ontologies.

Corpus-based approaches: For corpus based techniques, distributional semantics like PMI on Web-based corpora provide poor results [22]. This has already been discussed in section 2 (and [22]), and can be attributed to the fact that modern Web search engines API's are too limited and only return approximatively word counts for performance reasons. Therefore, we conclude that many interesting experiments based on Web search technologies cannot be replicated nowadays (e.g., [46], but also the studies of Sahami [21] and Bollegala [20] which reported good results in their time). But still, when using PMI measures restricted to a local Wikipedia corpus or the Factiva Corpus, the results are at least mediocre (correlation 0.4 for RG65).

Techniques based on more sophisticated implementations of distributional semantics using a local corpus all show very solid results, with correlation values around 0.6 for the full WS353 test, and around 0.8 for RG65. Especially sparse context-window (CW) representations show good results for similarity-based test sets (RG65, MC30, WS353-S). However, these experiments use an extremely large web-corpus, which consists of $1.6E^{12}$ words. Obtaining and processing such a large corpus is very difficult. As the accuracy of corpus-based techniques increases with the size of the corpus [23], one must assume that the approach will perform less well on a smaller corpus (as long as it is similarly constructed). Dense vector representations as for example SVD or Shifted PMI (SPMI) show comparably strong results even when using the significantly smaller Wikipedia corpus with $1.5E^{9}$ words.

We also performed experiments with neural word embeddings (SGNS [45], using common default setting of 300 dimensions, a window size of 5, and dropping all words which occur less than 5 times overall) on the Wikipedia corpus, which shows comparable performance (0.61 for WS353). During construction of these embeddings, no preprocessing was performed. Therefore, each word is treated as a unique token (i.e., as a result, "horse" and "horses" are treated as different concepts which are highly similar to each other). Word embeddings using the GLOVE [29] algorithm perform with comparable results.

We assumed that the performance of word embeddings might improve when pre-processing techniques like stemming would be applied (which would map dog and dogs to the same token). However, the results when using a stemmed corpus are inconclusive: while they slightly differ

Table 1: Evaluation Results on the MC39, RG65, and WS353 corpus

	Corpus / KB used	*Source*	*MC30* [39]	*RG65* [38]	*WS353* ***S*** [23]	*WS353* [40]	*WS353* ***R*** [23]	*MEN* [41]	*SIMLEX* [42]
Knowledge-based									
Shortest Path	WordNet		0.75	0.77	0.59	*0.37*	*0.07*	*0.36*	0.45
Leacock-Chod. [10]	WordNet		0.79	0.84	0.64	*0.32*	*0.01*	*0.34*	0.29
Resnik [12]	WordNet		0.81	0.83	**0.67**	*0.36*	*0.03*	*0.43*	0.35
Wu & Palmer	WordNet		0.76	0.78	0.61	*0.28*	*0.00*	*0.35*	0.32
Pirrò & Secco [48]	WordNet		0.83	0.83	0.46	*0.33*	*0.06*	*0.30*	0.22
Jiang & Conrad [14]	WordNet		**0.84**	0.83	0.38	*0.30*	*0.06*	*0.28*	0.20
Lin [13]	WordNet		0.83	**0.85**	0.63	*0.32*	*0.01*	*0.40*	0.39
FaITH [16]	WordNet		0.81	**0.85**	0.57	**0.39**	**0.38**	0.37	0.32
Simple Distributional									
PMI Bing	Web	[22]	0.08	0.11		0.00			
PMI Google	Web	[22]	0.05	-0.06		0.10			
PMI	Wikipedia	[22]	0.50	0.40		0.25			
PMI	Factiva	[22]	0.31	0.43		0.31			
Sahami [21]	WebSnippets	[23]	0.62						
SemSim [20]	WebSnippets	[20]	0.81		0.77	**0.67**	0.59		
Context Windows [23]	WebCorpus	[23]	**0.88**	**0.83**	0.77	0.60	0.46		
Bag of Words [23]	WebCorpus	[23]	0.85	0.68	0.70	0.65	0.62		
Dense Vectors									
LSA [24]	Tasa	[24]	0.73	0.65		0.52			
SGNS [45]	GoogleNews		0.79	0.77	0.78	0.64	0.64	**0.73**	0.44
SGNS [45]	Wikipedia		0.58	0.63	0.70	0.60	0.50	0.69	0,34
SGNS [45]	Wikip.Stemm.		0.60	0.66	0.67	0.61	0.55	0,66	0,31
SVD [37] *(k=5)*	Wikipedia	[37]				**0.69**		**0.73**	
SPMI [37] *(k=5)*	Wikipedia	[37]				0.61		0.69	
GLOVE [29]	GoogleNews		0.84	0.75	**0.80**	0.65	**0.68**	0.65	0.55
RNNenc [31]	Wikipedia EN-DE	[33]				0.61		0.62	0.50

from the results when using unprocessed text, there is no clear trend visible. However, when training a neural word embedding on the smaller GoogleNews corpus ($1.0E^9$ words), results actually improve. This is likely due to the fact that large parts of the Wikipedia corpus do not actively contribute to the performance with respect to the challenges found in WS353, as they mostly cover "everyday" words. Therefore, Wikipedia pages on specialized topics won't help a lot when determining similarity between words in common language. In contrast, the smaller GoogleNews corpus covers a more common vocabulary with improved depth. This experiment clearly shows the dependency of the result quality on the choice of a corpus.

While modern techniques like SGNS, SVD, or GLOVE show good results on corpora like WS353 and MEN, they have a weakness which is currently hard to highlight due to a lack of good reference datasets. Basically, algorithms based on word vector representations do not explicitly measure similarity or relatedness, but a mix of both measures (i.e. SGNS assigns high values to both [horse, harness] which is related and [horse, donkey] which is similar). A challenge for current research is designing algorithms which can clearly distinguish between both concepts, which is hard for modern corpus-based techniques and would also require more elaborate reference datasets which specifically pay attention to the problem. Here. SIMLEX-999 is a good first step in that direction, but focuses mostly on similarity: in contrast to the other corpora, it does contain some pairs which are highly related but not similar, and rates them with a low reference score. This explains the comparably bad performance of corpus-based approaches on SIMLEX as they will assign a high value for such pairs. A good future algorithm should be able to distinguish between related-and-similar, related-and-not-similar, and not-related-nor-similar.

Currently, there are some assumptions on how to improve distributional models towards that goal [42] (but which will require further research): Models that learn from syntactic or dependency relations seem to perform better with respect to similarity (as shown by RNNenc [31], which is a model learned from a bilingual corpus for machine translation heavily emphasizing grammar), while those learning from running text or bag-of-words seem to handle relatedness better [37]. Also, larger context windows seem to favor better performance with respect to relatedness, and a smaller context windows favor similarity [47]. Combining these assumptions with other techniques (like merging distributional techniques and knowledge-based techniques) may lead to strong algorithms which can deal with the similarity-relatedness problem.

4. Summary & Outlook

In this paper, we discussed a wide variety of techniques for measuring semantic similarity and relatedness. Many traditional approaches rely on knowledge bases like ontologies or taxonomies, which limits them in several ways: for most domains, no suitable ontologies are available, and the construction of an ontology requires tremendous efforts. Furthermore, even when such a knowledge base is available, they are often not complete enough to properly support reliable similarity measures, while relatedness measurements are usually poorly supported or not at all. Especially in quickly changing domains like e-commerce, this is a major concern.

Therefore, we favor corpus-based approaches which can approximate similarity and relatedness by analyzing a large natural language corpus. Such approaches usually rely on the distributional hypothesis, and can therefore not distinguish between similarity and relatedness. However, they quickly adapt to changing semantics and work easily with special domains as long as there is enough natural text available. Also, they require little manual effort or tuning. Among corpus-based approaches, dense vector representations showed very good performance, even though they require longer computation times for creating the vector representations. Especially neural word embeddings, a set of new neural network-based techniques which currently rise in popularity, look promising. While neural embeddings perform similarly well as other dense vector techniques with respect to word similarity tasks evaluated in this paper, they have promising additional properties which invite future research (i.e., for similar effort, neural embeddings provide comparable or slightly better performance with respect to similarity and relatedness, but also offer more features and semantics for solving other tasks).

Especially, the semantics of neural embeddings seem to go far beyond simple similarity as also the more complex concept of relational similarity can be covered (e.g., the relation between Paris and France is similar to the one between London and Great Britain). This also allows for some limited similarity arithmetic like "Paris"-"France"+"Great Britain"="London". However, these features are not yet well understood, and it is unclear to which extend the relational similarity semantics of neural embeddings can be exploited.

Therefore, in future research, focus should be on exploring the limits of the semantics of neural word embeddings in several applications and use cases. This work can significantly advance the field of analogical reasoning, which is a central building block of natural user interfaces (e.g., intelligent personal assistants, and queries like "I am looking for a holiday location like Hawai'i, but in Asia.") A better understanding of neural word embeddings will also boost other fields as well, like ontology alignment, ontology construction, or knowledge management in scientific digital libraries (e.g., automatic construction of concept maps or unsupervised categorization systems).

[References]

[1] A. Tversky, "Features of similarity," *Psychol. Rev.*, vol. 84, no. 4, pp. 327–352, 1977.

[2] C. Lofi and C. Nieke, "Exploiting Perceptual Similarity: Privacy-Preserving Cooperative Query Personalization," in *Int. Conf. on Web Information System Engineering (WISE)*, 2014.

[3] Y. Matsuo, J. Mori, M. Hamasaki, T. Nishimura, H. Takeda, K. Hasida, and M. Ishizuka., "POLYPHONET: an advanced social network extraction system from the web," *Web Semant.*

Sci. Serv. Agents World Wide Web, vol. 5, no. 4, pp. 262–278, 2007.

[4] D. McCarthy, R. Koeling, J. Weeds, and J. Carroll, "Finding predominant word senses in untagged text," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2004.

[5] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi, "Personal Name Resolution Crossover Documents by a Semantics-Based Approach," *IEICE Trans. Inf. Syst.*, vol. 89-D, no. 2, pp. 825–836, 2006.

[6] G. A. Miller, "WordNet: a lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[7] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *SIGMOD Int. Conf. on Management of Data*, 2008.

[8] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "DBpedia: A Nucleus for a Web of Open Data," *Semant. Web*, vol. 4825, pp. 722–735, 2007.

[9] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *16th international conference on World Wide Web (WWW)*, 2007.

[10] C. Leacock and M. Chodorow, "Combining Local Context and Wordnet Similarity for Word Sense Identification," in *WordNet: An Electronic Lexical Database*, MIT Press, 1998, pp. 265–283.

[11] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 1994.

[12] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in *Int. Joint Conference on AI (IJCAI)*, 1995.

[13] D. Lin, "An information-theoretic definition of similarity," in *Int. Conf. on Machine Learning (ICML)*, 1998.

[14] J. J. Jiang and D. W. Conrath, "Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy," in *Conference on Linguistics and Speec Processing (ROCLING)*, 1997.

[15] N. Seco, T. Veale, and J. Hayes, "An intrinsic information content metric for semantic similarity in WordNet," in *European Conference on Artificial Intelligence (ECAI)*, 2004.

[16] G. Pirrò and J. Euzenat, "A Feature and Information Theoretic Framework for Semantic Similarity and Relatedness," in *Int. Semantic Web Conference (ISWC)*, 2010.

[17] Z. Harris, "Distributional Structure," *Word*, vol. 10, pp. 146–162, 1954.

[18] K. Church and P. Hanks, "Word association norms, mutual information, and lexicography," *J. Comput. Linguist.*, vol. 16, pp. 22–29, 1990.

[19] P. D. Turney, "Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL," *Lect. Notes Comput. Sci.*, vol. 2167, pp. 491–502, 2001.

[20] D. Bollegala, Y. Matsuo, and M. Ishizuka, "Measuring semantic similarity between words using web search engines," in *Int. Conf. on World Wide Web (WWW)*, 2007.

[21] M. Sahami and T. D. Heilman, "A web-based kernel function for measuring the similarity of short text snippets," in *Int. Conf. on World Wide Web (WWW)*, 2006.

[22] A. Panchenko, "A Study of Heterogeneous Similarity Measures for Semantic Relation Extraction," in *JEP-TALN-RRECITAL*, 2012.

[23] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Pasca, and A. Soroa, "A study on similarity and relatedness using distributional and WordNet-based approaches," in *Ann. Conf. of the North American Chapter pf the Association for Computational Linguistics (NAACL)*, 2009.

[24] D. Ştefănescu, R. Banjade, and V. Rus, "Latent Semantic Analysis Models on Wikipedia and TASA," in *Language Resources Evaluation Conference (LREC)*, 2014.

[25] T. K. Landauer and S. T. Dumais, "A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge," *Psychol. Rev.*, vol. 104, no. 2, pp. 211–240, 1997.

[26] Y. Bengio and J.-S. Senécal, "Quick training of probabilistic neural nets by importance sampling," in *AISTATS Conference*, 2003.

[27] A. Mnih and G. E. Hinton, "A scalable hierarchical distributed language model," *Adv. Neural Inf. Process. Syst.*, pp. 1081–1088, 2009.

[28] T. Mikolov, W. Yih, and G. Zweig, "Linguistic Regularities in Continuous Space Word Representations," in *Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language (NAACL-HLT)*, 2013.

[29] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Conf. on Empirical Methods on Natural Language Processing (EMNLP)*, 2014.

[30] N. Kalchbrenner and P. Blunsom, "Recurrent Continuous Translation Models," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

[31] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[32] F. Hill, K. Cho, S. Jean, C. Devin, and Y. Bengio, "Not All Neural Embeddings are Born Equal," in *NIPS Workshop on Learning Semantics*, 2014.

[33] F. Hill, K. Cho, S. Jean, C. Devin, and Y. Bengio, "Embedding Word Similarity with Neural Machine Translation," in *Int. Conf. on Learning Representations (ICLR)*, 2015.

[34] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language

processing (almost) from scratch," *J. Mach. Learn. Res.*, no. 12, pp. 2493–2537, 2011.

[35] W. Y. Zou, R. Socher, D. M. Cer, and C. D. Manning, "Bilingual Word Embeddings for Phrase-Based Machine Translation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

[36] T. Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., & Mikolov, "Devise: A deep visual-semantic embedding model," *Adv. Neural Inf. Process. Syst.*, pp. 2121–2129, 2013.

[37] O. Levy and Y. Goldberg, "Neural Word Embedding as Implicit Matrix Factorization," in *Conf of the Neural Information Processing Foundation (NIPS)*, 2014.

[38] H. Rubenstein and J. B. Goodenough, "Contextual correlates of synonymy," *Commun. ACM*, vol. 8, no. 10, pp. 327–633, 1965.

[39] G. A. Millera and W. G. Charles, "Contextual correlates of semantic similarity," *Lang. Cogn. Process.*, vol. 6, no. 1, pp. 1–28, 1991.

[40] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin, "Placing search in context: the concept revisited," in *Int. Conf. on World Wide Web (WWW)*, 2001.

[41] E. Bruni, N. K. Tran, and M. Baroni, "Multimodal Distributional Semantics," *J. Artif. Intell. Res.*, vol. 49, pp. 1–47, 2014.

[42] F. Hill, R. Reichart, and A. Korhonen, "SimLex-999: Evaluating Semantic Models with (Genuine) Similarity Estimation.," *Prepr. Publ. arXiv. arXiv14083456*, vol. 2014.

[43] D. Gentner, "On relational meaning: The acquisition of verb meaning," *Child Dev.*, pp. 988–998, 1978.

[44] S. Bird, "NLTK: the natural language toolkit," in *Int. Conf. on Computation Linguistics (COLING)*, 2006.

[45] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," *Adv. Neural Inf. Process. Syst.*, pp. 3111–3119, 2013.

[46] D. T. Bollegala, Y. Matsuo, and M. Ishizuka, "Measuring the similarity between implicit semantic relations from the web," in *Int. Conf. on World Wide Web (WWW)*, 2009.

[47] D. Kiela and S. Clark, "A systematic study of semantic vector space model parameters," in *Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, 2014.

[48] G. Pirró, "A semantic similarity metric combining features and intrinsic information content," *Data Knowl. Eng.*, vol. 68, no. 11, pp. 1289–1308, 2009.

Christoph LOFI is currently a researcher at Institute for Information Systems (IfIS) at Technische Universität Braunschweig. His research focuses on human-centered information system queries, therefore bridging different fields like database research, natural language processing, or crowdsourcing. Between 2012 and 2014, he was a post-doctoral research fellow at National Institute of Informatics in Tokyo, Japan. Before, he was with Technische Universität Braunschweig since 2008, where he received his PhD degree in early 2011. He received is M.Sc. degree in computer science in 2005 by Technische Universität Kaiserslautern, Germany, and started his PhD studies at L3S Research Centre, Hannover, in early 2006.