Learning Mechanisms on Semantic Service Descriptions for Automated Action Selection

Johannes Fähndrich, Sabine Weber, Sebastian Ahrndt, and Sahin Albayrak

DAI-Laboratory of the Technische Universität Berlin
Department of Electrical Engineering and Computer Science
Ernst-Reuter-Platz 7, 10587 Berlin, Germany
johannes.faehndrich@dai-labor.de (Corresponding author)

Abstract. The capability to identify the sense of polysemic words, *i.e.* words that have multiple meanings, is an essential part of intelligent systems, e.g. when updating an agent's beliefs during conversations. This process is also named Word Sense Disambiguation (WSD) and is approached by applying semantic similarity measures. Within this work, we present an algorithm to create such a semantic similarity measure using marker passing, that: (1) generates a semantic network out of a semantic service description, (2) sends markers through the networks to tag sub-graphs that are of relevance, and (3) uses these markers to create a semantic similarity measure. We will discuss the properties of the algorithm, elaborate its performance with different part of speech, and discuss the lifted properties for the algorithm to be used in WSD. To evaluate our approach, we compare it to state-of-the-art measures using the *Rubinstein1965* and *WordSim353* dataset. It is shown, that our approach outperforms these state-of-the-art measures and, further, is able to adapt the predictions to the contextual information.

1 Introduction

In a recent work [25], it was argued that the applicability of agent technologies to real world problems is limited. Part of the conclusion, that was drawn by the authors, is that interoperability between agent-based systems and other more used approaches such as service oriented architectures (SOA) [10] is a key factor for the acceptance of our techniques. Among others, the authors inferred that capability descriptions, matching techniques (e.g., ontology matching), and planning approaches provide significant challenges when facing this interoperability.

The integration of agent-oriented and service-oriented technologies provides manifold advantages. One reason for that is the wide spread acceptance of web services. Another one is the successful standardisation process. Bridging the gaps between AOSE and SOA would enable agents to use a vast amount of web services as additional sensors and actors. In fact, the agent-community identified this opportunity long ago. However, many approaches use a static combination of web services that is determined during design time [25]. *But is it not that* *autonomous agents should decide which capabilities to use during runtime?* To enable this, there is the need to extend the capability of available planning techniques towards the ability to use semantic service descriptions. One key-factor in this circumstance is the ability to identify the meaning of a description, which is in the focus of our research.

This paper approaches the problem of ontology matching on service descriptions and introduces a learning mechanism for new concepts. If for example the goal of an agent states "a meeting at noon" and a service provides "parking estimates for midday parking" the agent needs to identify noon with midday. In this way all concepts describing a service are analysed and new concepts are integrated in to the beliefs of the agent. Here the first task is to identify the sense of a word. Therefore, we reduce the problem to the one named Word Sense Disambiguation (WSD), which is the process of identifying the sense of polysemic words, that is words which have more then one meaning. Research on this area so far has identified WSD as a main problem of languages and language understanding. In [2], an overview about the area and available approaches is given. WSD itself is a sub-area of natural language understanding and an AI-complete problem [18, 44]. We are in this area, as in state-of-the-art ontology matching approaches words only have one sense, which simplifies the problem towards the lookup of the meaning of a word in a dictionary. However, identifying the sense of a word in a given context is an essential basis for language understanding and is done by providing a measure of the semantic similarity of words.

> For the sake of an example, we refer to the beliefs of an agent. By integrating new concepts into its beliefs the agents could be able to extend the actions available to it by becoming able to search, identify, and use new services. Here our example is that an agent knows the concept "noon" and learns the new concept "midday". This integration is done by assigning new concepts to known concepts. The first hurdle an agent has to over come is the ability to find out if the new concept represents something it already knows and in which relation the new concept stands to older once. This establishes a common ground, which allows further communication with other agents or which allows a further analysis of a given semantic service description.

Representing meaning as a graph is one of the two ways AI, cognition and linguistic researchers think about meaning (so-called connectionist view). Logicians and formal representation of meaning, on the other side, include the symbolic representation, where description logics is used to describe the language and the meaning of symbols and their references. This *neats* vs. *scruffy* discussion is going on for the last 40 years [30]. The approach proposed here combines those two views by integrating symbolic information into a connectionist approach.

For the remainder of this work, we first look at the state-of-the-art of semantic similarity measures (Section 2). Then we describe how knowledge is represented in a graph structure that serves as foundation for the representation of meaning.

Upon this graph marker passing[1] is used to create the dynamic part of meaning representing thoughts [9]. The marker passing algorithm uses node and edge interpretation to guide its markers. The node and edge interpretation models the symbolic influence of certain concepts [7]. To evaluate the resulting artificial representation of meaning an experiment is presented in which the parameters of the marker passing are evaluated through finding semantic similarities between concepts [31]. We finally compare the reached results to state-of-the-art measures using the *Rubinstein1965* and *WordSim353* dataset. It is shown, that our approach outperforms these state-of-the-art measures and, further, is able to adapt the predictions to the contextual information.

Fig. 1 depicts our abstract approach, from the input of two concepts, which are decomposed into a semantic graph (Section 3). Then we use marker passing to identify relevant sub-graphs (Section 3.1) and describe the parameters use to interpretate the marker information (Section 3.2). The whole concept is used to create a semantic similarity measure that is experimentally evaluated (Section 4).

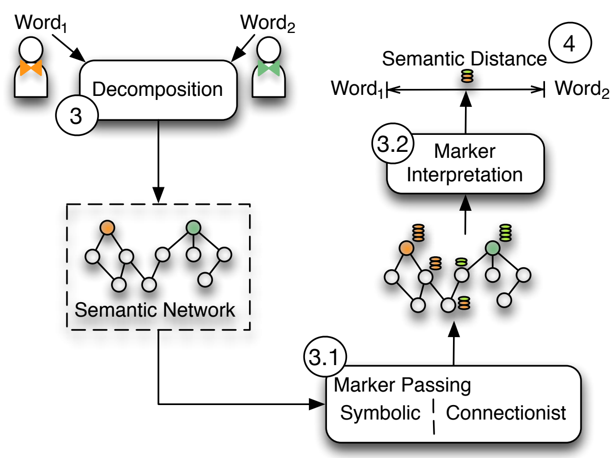

Fig. 1. Overview of the algorithm to measure semantic similarity of two concepts.

2 Related Work

The research literature on word similarity metrics covers a wide range of approaches ranging from simple thesaurus based approaches (*cf.* [19]) to complex neuronal networks (*cf.* [28]). These approaches can be classified into groups based on the data structure that is applied. Most commonly the differentiation is made between *knowledge-based* and *corpus-based* work [27]. *Zhang et al.* [49] employ the same division in their survey, further subdividing the domain of knowledge-based approaches into *taxonomies* and *ontologies*. Taxonomies are organised

[1] Sometimes referred to as Activation Spreading or Token Passing

by the generalisation-specialisation relationship, while ontologies are taxonomic structures enriched with other semantic relationships. Based on this classification we selected three state-of-the-art approaches. Each one being a representative for one of the groups, that will be introduced in more detail next and used as baseline during our evaluation. We selected the *Electronic Lexical Knowledge Base* (taxonomy-based), the *Bidirectional One-Step* approach (ontology-based), and *Word2Vec* (corpus-based) approach as typical representatives of their group. *Lastra-Diaz et al.* [21] has not been selected because it mixes corpus based and ontology based approaches. The approach of *Mikolov* [28] has been selected as it provides a better performance than the algorithm of *Baroni* [4]. Bringing the categories and approaches together, Fig. 2 shows the development of the most prominent word similarity metrics on a time scale, starting with the approach of Rada et al. [34] and finishing with the approach of Lastra-Diaz et al. [21].

We compare our approach to all measures within the same performance range of the state-of-the-art according to the surveys of *Pilehvar et al.* [33], *Lastra-Díaz, García-Serrano* [21], and *Zesch and Grevych* [48]. Excluding the approach of Yang [45] (denoted as YP05) because it has been performed on a subset of the dataset.

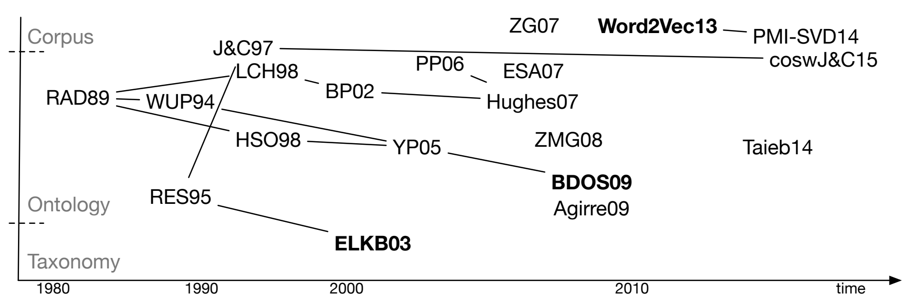

Fig. 2. Overview of the State-of-the-Art of semantic similarity approaches. These are the same approaches compared to in the evaluation in Table 2

2.1 Taxonomy-based: Electronic Lexical Knowledge Base (ELKB)

One data base used to find semantic similarities between words is the digitised version of *Roget's Thesaurus*. The structure of the thesaurus is similar to a taxonomy. It is a tree-like network, in which most connections are of kind parent-child. The best performing algorithm on this structure was introduced by *Jarmasz et al.* [19] in 2003 and named ***E****lectronic* ***L****exical* ***K****nowledge* ***B****ase*. The semantic similarity between two words is the length of the shortest path found between the two concepts. That is to say, if r_1 and r_2 are the sets of references for the words, then the similarity (sim) between the words w_1 and w_2 is calculated as:

$sim(w_1, w_2) = 16 - min_{distance}(r_1, r_2)$, where 16 is the maximum height of the tree. Later, the authors compared their approach to other state of the art approaches like Hirst-St.Onge, Jiang-Conrath, Leacock-Chororow, Lin and Resnik to show that their approach yields the best results [16, 22, 24, 36].

2.2 Ontology-based: Bidirectional One-Step (BDOS)

Another data base used to find semantic similarity between words is WordNet[2], a lexical data base developed at Princeton University. Similar to Roget's Thesaurus words in WordNet are connected trough their relations to each other, creating a graph. Most approaches using this structure for the measurement of semantic similarities work similar to those used on Roget's Thesaurus. Semantic similarity between two words is derived from the distance between their representations in the graph. Differences between approaches consist in which links are used and how the links are counted.

One of the most powerful of these approaches is the ***Bid**irectional **O**ne-**S**tep* algorithm proposed by *Chen et al.* [8] in 2009. It utilises the hypernym, hyponym and synonym relations and expands up and down the hierarchy simultaneously. In each step of the algorithm adjacent nodes are added to the respective sets and the sets are compared. If no overlap is found the next adjacent nodes are added. If an overlap is found the number of steps in the algorithm is the length of the shortest path. *Chen et al.* compare their results to the results achieved by *Yang et al.* whose metric was closer to human judgement the the state of the art in 2009.

2.3 Corpus-based: Word2Vec

A third way to obtain knowledge about the human perception of semantic similarity refrains from the use of predefined networks, but utilises the vast amounts of language data produced by humans. The different ways to use large corpora of language are based upon the assumption that two words are similar when they can be used interchangeably without altering the meaning of a statement, *i.e.* similar words are used in similar contexts. When these contexts of words are represented in a matrix, the different vectors of the matrix can be compared to each other. The similarity of the vector representation of two words can then be seen as the similarity of these two words.

One prominent approach in this group of algorithms is proposed by *Mikolov et al.* [28]. The authors use a neuronal network to create the vector representation of a large text corpus. The network is trained to guess the words in the immediate context. The semantic similarity between two words is then calculated as the cosine similarity of the two vectors representing them. In the Microsoft Research Sentence Completion Challenge [3] this approach achieved the highest accuracy of all competing methods.

[2] http://wordnet.princeton.edu/wordnet/

[3] `http://research.microsoft.com/apps/pubs/?id=157031`

In conclusion, all these approaches are not context dependent and, thus, lack the ability to select a semantic similarity specific to a given context. This context can be the knowledge of the agent or some additional information which is specific to the time of execution.

3 Creating a Semantic Similarity Measures

To create a semantic similarity measure our approach utilises semantic decomposition [11] to extract relevant information from a set of heterogeneous information sources. Therefore we extend the approach of [20]. The semantic graph is enriched with all available information from the sources: *WordNet*[4], *Wiktionary*[5], *Wikidata*[6] and *BabelNET*[7]. Here the graph is created by lexical decomposition that breaks each concept down semantically until a set of semantic primes is reached [37, 12]. The primes are taken from the theory of the Natural Semantic Metalanguage (NSM) [15, 41]. Here the theory states that all natural languages have a semantic core consisting of 65 semantic primes in which each word can be decomposed. In the created graph, nodes and edges are no longer disjoint since a word can appear as a relation or as a concept. With this semantic decomposition the beliefs of an agent are used as initial knowledge for building a semantic graph representing the believed facts of the agent. The complete decomposition is shown in Alg. 1.

An example result of the algorithm 1 is simplified in Figure 3.

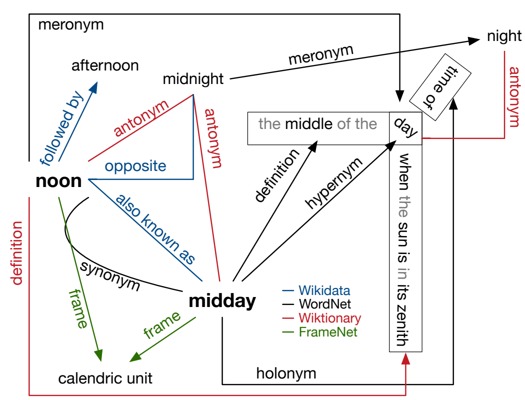

Fig. 3. Example graph for the words midday and noon.

In Figure 3 the relations form different datasource are depicted in different colours. If edges are symmetric they are shown without direction and mot of the

[4] https://wordnet.princeton.edu/
[5] https://en.wiktionary.org/
[6] https://www.wikidata.org
[7] http://babelnet.org/

Algorithm 1 A decompositional algorithm

Name: Decompose **Input:** concept, PRIMES **Output:** Concept

```
concept ← Normalisation(concept)
Relations ← getRelations(concept)
Definitions ← lookUpDefinitions(concept)
if concept ∈ PRIMES then return concept
end if
for all r ∈ Relations do
    if r ∈ PRIMES then
        AddRelation(Concept, r, getPrimOfConcept(r))
    else
        AddRelation(concept, r, r.target)
        decompose(r,PRIMES)
        decompose(r.target, PRIMES)
    end if
end for
for all definition in Definitions do
    for all def in definition do
        AddRelation(Concept, "definition", def)
        if def in PRIMES then
            continue
        else
            decompose(def,PRIMES)
        end if
    end for
end forreturn concept
```

edges have been removed to increase readability. Building a bigger network in this way allows us to broaden the context in which we can interpret concepts. A semantic graph is built for each new concept, extending the beliefs of the agent towards the meaning of that concepts. As a result, the semantic graph contains additional linguistic relationships like antonyms, synonyms, hyponyms or hypernyms.

As introduced in Section 1, one main hurdle is to link new concepts to existing ones. Using the algorithm for different concepts, enables us to merge the resulting semantic graph pairwise. The outcome is used in the next step, which is the marker passing. Here the task is to compare two concepts. The marker passing uses these concepts as starting points to activate relevant sub-graphs. This algorithm will be explained next.

3.1 Marker Passing in Context

The marker passing algorithm is distinguished by four phases. These are depicted in Fig. 4: (1) the *pre-processing* for preparing (priming) the graph; (2) the *selection of the pulse size*, which defines which nodes will pass information within the pulse; (3) the *pulse* itself, where all spearing nodes are activated and pass on their markers (each pulse consists of a *activation step* for each active node); and (4) the *post-processing* step, were the results i.e. are normalised. The input parameters for the marker-passing algorithm are the node interpretations, the termination condition and the underlying graph.

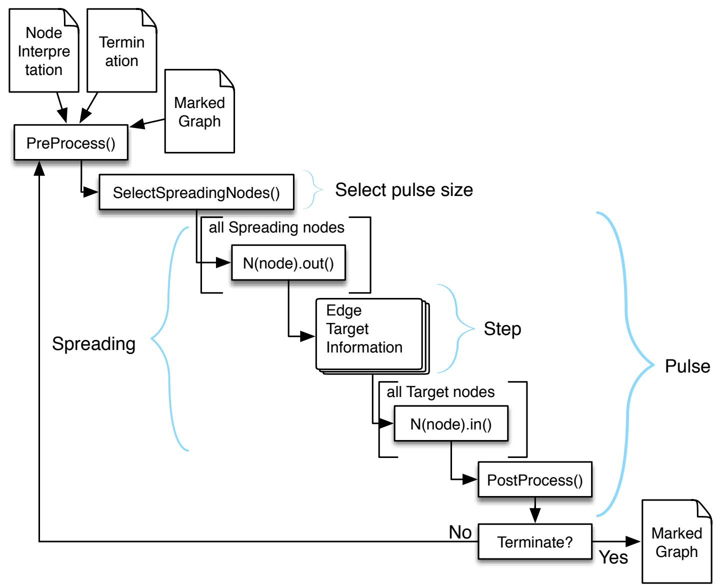

Fig. 4. Overview of the algorithm to measure semantic similarity of two concepts.

In the applied type of marker passing the nodes and edges of the underlying graph do not have to be disjoint. This has the benefit, that if needed a edge can be interpreted as a node and spread to other edges.

For our example, we put start markers on the two nodes "noon" and "midday". Each markers hods information from which concept it started and how much activation it carries. Staring from there "noon" passes markers i.e. in the first puls to each concept of its definition "time of day where the sun is in its zenith". Midday on the other hand activated its hypernym "day" where now markers from both concept meet. In one plus the active concept passes markers to all its outgoing edges depending on its interpretation function and the interpretation of the edge.

Similar to Thiel and Berthold [40] the marker passing can be abstracted to a graph $G = (V, E, w)$ where the nodes (V) and the edges (E) represent the pulse of all spreading nodes and the edges which are selected to spread activation. In contrast to Thiel and Berthol this G does not represent the entire semantic network but only one puls. This is because the activation of each puls depends on the puls before and thus the weights might change in each puls. The activation step can be formalised as follows:

$$\hat{a}^t = \sum_{u \in V} u.out(a^{t-1}(u), e) \tag{1}$$

Where $a^{t-1}(u)$ is the activation of node u at puls $t-1$ with $t > 0$ and $\hat{a}^t(v)$ being the activation step, where all active nodes pass their markers regarding their out-function. Now the markers are passt to their destination and are interpreted by the in-function:

$$\hat{a}^*_t(v) = \sum_{\forall u \in V : \exists e(u,v) \in E} v.in(\hat{a}^t, v, e) \tag{2}$$

where $\hat{a}^*_t(v)$ represents the activation of the node after all the markers passed over the edges connected to v are incorporated by the in-function of the node v into its current activation.

This activation can then be used to calculate the semantic similarity in the following way:

$$d_{sim}(u,v) = \frac{\sum_{t=0}^{t_{max}} \sum_{x \in V} \Phi(\hat{a}^*_t(x))}{\sum_{\forall w \in V} a^0(w)} \tag{3}$$

where

$$\Phi(\hat{a}^*_t(x)) = \begin{cases} \hat{a}^*_t(x) & \text{, if } a^0(u), a^0(v) \subseteq \hat{a}^*_t(x) \\ \emptyset & \text{, else} \end{cases} \tag{4}$$

$\Phi(\hat{a}^*_t(x))$ filters all nodes which have not been activated by at least two of the start activations. In this way, if we activate two concepts (in our example "noon" and "midday") at the beginnen, this set contains all nodes which have been activated by markers of both concepts. This is normalised by the amount of start activation to obtain the semantic distance. t_{max} is defined by the termination condition of the marker passing.

How the marker passing is configured e.g. how the weights of the edges are selected or how the in- and our-function are defined, is subject to the next section.

3.2 Parameters of the marker passing to create a semantic similarity measure

The novelty in the marker passing algorithm used is the node and edge interpretation function denoted by $N()$ in Fig. 4. This interpretation functions allow each node and edge to react distinctively to markers passed from ($N.out$) or to ($N.in$) them. With this symbolic information each node can interpret each marker. The behaviour of each node is specified within these two functions, whereas the in-function specifies what to do with incoming markers and the out-function specifies how markers are passed to the nodes connected to this node with edges. A spreading step defines which information is passed during one pulse. The size of the pulse can be chosen by the *SelectSpreadingNodes* function, which selects those nodes which activate during the next pulse from the set of *active nodes*. Active nodes are those nodes which reach some predefined threshold. This threshold and the in- and out-function is part of the node interpretation. After each pulse a given termination condition is checked. This termination condition prevent the markers to be equally distributed over the network [5].

In the experiment carried out two words are given without context, part of speech information or word sense. In our approach the two words are given to the decomposition which creates a semantic graph based on the information sources used by the decomposition. The result is a semantic graph for each word. These graphs are without contextual information. To set an example: If a word like "bank" is given, one has to decide if this word is part of the common ground. Here, the sending agent could talk about a financial institute, whereas the receiving agent could talk about a geological formation or a sitting opportunity.[8].

As information on the marker we chose a numeric activation level and a marker "origin" encoding the starting node of the marker. We chose this symbolic information since it seams minimalistic without reducing the marker passing to a triviality.[9] The activation level is used to measure the distance to the origin node. We use the "origin" of the marker to react to cases in which a node is activated by at least two different concepts.

The underlying graph consists of vertexes, where each vertex represents one concept of the decomposition. The edges of the graph represent the relations of the decomposition. Each edge has a edge type corresponding to its semantic relation in the decomposition and a numerical weight — for example, an antonym relation creates a antonym edge with a specific weight. All semantic relations known to the decomposition are weighted edges. The following edge types have been specified:

- **Semantic relations** are relations out of the dictionaries used during the decomposition. Here we take the following relations into account: synonyms (semantically similar in some contexts), antonyms (the opposite of something), hyponyms (specialisation), hypernyms (generalisation) and meronyms (is part of relations).
- **Others** are relations that are not specified in the dictionaries used in the decomposition but are taken from the domain ontology of the agent. Here relations like *"is uncle of"* or *"is owned by"* could be introduced into the decomposition. There edges in the resulting graph are weighted with 1.0.

[8] Even with contextual information such concepts are not always easy to identify, as shown by Bar-Hillel [3], e.g. *"The box is in the pen"*

[9] With less symbolic information the marker passing becomes activation spreading, which in the special case of artificial neuronal networks is subject to research.

Similar to the edge type, the graph consists of different types of nodes. The following node types have been specified:

- **Concept nodes** represent a concept and hold the markers passed to it. The threshold of a concept node is reached if one of the marker origin reaches the numeric activation threshold τ.
- **Stop word nodes** represent words which are ignored. Those are taken form natural language processing theory [42]. Those nodes can not be activated.
- **Prime nodes** represent semantic primes from the NSM, which act as leave nodes and collect markers without passing them.

Furthermore, there are infinite specialisation methods for the marker passing. We have made the following design decisions to build our semantic similarity measure:

- There is neither a pre- nor a post-processing step.
- Start Marker are put on the two initially decomposed concepts. The start marker declares the origin of the marker. All start markers specifies a equal activation level.
- The threshold $\tau(n)$ is a selected numeric value, that is checked against the total activation of the node.
- The in-function of a node collects the markers in the "marker passing step", which the current node is the target of. Markers of all edges are sorted regarding their origin. Furthermore, the in-function sums-up the activation of all origins to update the activation level of the node.
- The Pulse size was selected in a way that all active nodes are spreading nodes instead of having, e.g., each node activated in an own pulse. This has been done since the activation is collected additively in our in-function and does not decay (reduced over time) in a node.
- The out-function propagates markers with the total amount of activation to all edges weighted with the same edge weight. Additionally the markers at the moment of activation are held in an activation history for later analysis. Thus, after an activation the node has no markers of any origin.
- The termination condition is set to the maximum step count one marker can achieve in the graph.

After the marker passing step, the graph is activated and must be interpreted. In Fig. 1 this step is called marker interpretation. The idea is that closer concepts will have more markers on the same vertexes compared to concepts at greater distances. The marker passing algorithm can create such an representation if configured in the right manner. Next we will introduce the configuration we used to create our results. The symbolic information is used in the following way: The marked graph is analysed for nodes which have passed markers of multiple origins. The activation of all markers that were passed to this node (independent of the origins) is summed up. To do so, the node history is used to look-up the total activation the node has experienced over time. Thus nodes which have a throughput of markers of multiple origins contribute more activation to the final marker count. Additionally the node type allows us to ignore stop words and stop the activation if a semantic prime is reached.

4 Evaluation

Evaluating is done on linguistic datasets where word similarity is measured. No ontology matching dataset is used, since ontology matching datasets manly focus on structural matching. This can be seen as facilitation of the semantic similarity task, where additional information is provided.

Datasets To evaluate the presented approaches we selected widely-used datasets. The most frequently used is the one published by Rubenstein and Goodenough 1965 (RG65) [38]. It consist of 65 noun pairs. The semantic similarity of these pairs were rated on a scale of zero to four by ten test subjects. Most papers dealing with metrics for semantic similarity use this dataset, which makes it suitable for the verification of claimed results and comparison between different metrics as presented in Section 4. Other datasets that were applied are the WordSimilarity-353 Test collection [13], the Miller and Charles dataset [29], the MTurk dataset [35], the MEN dataset [6], and the Stanford Rare Word Similarity dataset [26]. We selected these datasets to assure that the selection of words compared to each other is as big as possible. Moreover we wanted to make sure, that not only the similarity between nouns, but between all different word types is evaluated. Nevertheless, since most of the related work found is compared using the RG65 [38] dataset, we compare our results as well on this dataset. We normalised the dataset RG65 to contain similarity values between 0 and 1.

4.1 Parameter selection

The parameters selected in Table 1 depend on design decision during the creation of the marker passing. Here many combinations of parameters can yield the same or better results and we do not claim that our parameters yield the optimal result. However, later it can be seen that even without parameter optimisation our approach can keep up with the state-of-the-art.

To come up with the an estimate of the best value for the parameters, we have used an evolutionary algorithm. Here the individual's DNA of an initial population is made up of random values. The individual evaluation is done with a n-fold cross-validation using the RG65 [38] dataset with 65 folds. Evolution is done by mutating the best individual in all parameters randomly. One exception is the custom relations out of the domain ontology, which are weighted 1.0.

The best result was yield by the parameters shown in column 'best' of Table 1. Interesting for this result is that the parameters are interdependent. Having synonym edges with negative weights seams not intuitive. Also that the weights of antonym edges are less negative to the ones of synonyms. This can be explained with the occupancy frequency of the relation types. WordNet includes more synonym relations then antonym relations. Additionally the dataset RG65 consist solely of nouns, where antonyms are infrequent. The decomposition depth has been limited to 2 because of computational resources, which were not able to handle decomposition of deeper levels. The parameter selection is significant, since the results from one example learning run reaches correlation from 0.008

Table 1. Parameters of the marker passing approach used for the task of semantic similarity measurement.

Parameter	RG-65		
	min	max	best
startActivation	0	100	3.30
threshold	0	1	0.32
DefinitionLinkWeight	-1	1	0.25
synonymEdgeWeight	-1	1	-0.94
antonymEdgesWeight	-1	0	-0.11
hypernymEdgesWeight	-1	1	0.30
hyponymEdgesWeight	-1	1	0.11
terminationPulsCount	1	100	99
doubleActivationLimit	0	1	0.53
decompositionDepth	1	2	2

to 0.832 in 45.253 generations. Additionally the parameters are specially trained for the properties of the example dataset. Here the learning needs to be repeated for each new dataset with different properties.

4.2 Evaluation results

The experiment has been implemented in Java using the WordNet 3.1 and the MIT Java Wordnet Interface (JWI)[10]. For the Wiktionary implementation the Java-based Wiktionary Library (JWKTL)[11] has been used with a Wiktionary dump[12]. With the above introduced parameter selection, we were able to reach the results shown in Table 2. The other results are taken from [33, p. 116, Table 9] and from [21, p. 148, Table 4] and present the state-of-the-art in this experiment.

The spearman's ranked correlation coefficient ρ is an overall ranking measure. Together with the pearson's r Table 2 shows the performance of the state-of-the-art semantic similarity measures in comparison to each other.

Fig. 5 illustrates the reached results for the RG65 dataset, comparing our results to the three reference approaches we selected. Here, the x-axes represents the similarity of a word pair out of the RG65 dataset. The y-axis listed the 65 different word pairs ordered from semantically close concepts (synonyms in the best case) to concepts at greater semantic distances. Most of the time, the marker passing (MP, blue) underestimates the similarity in contrast to our three reference measures. The underestimation gets worse the smaller the semantic similarity is, until at the far end the measure overestimates the similarity. But we can see that the linear progression of the MP approach is closest to the human (green) guess of similarity. The best average performance of the other approaches is reached by the ELKB approach (red), which overestimates the similarity most

[10] `http://projects.csail.mit.edu/jwi/`
[11] `https://www.ukp.tu-darmstadt.de/software/jwktl/`
[12] `https://dumps.wikimedia.org/` downloaded on 2015.12.19

Table 2. Comparison of Spearman's ρ and Pearson's r correlation coefficients of different approaches with our approach. Anything but our results are taken from [33, p. 116, Table 9] and from [21, p. 148, Table 4].

Approach	RG-65	
	ρ	r
PP06[32]	0.62	0.58
ELKB[19]	0.65	–
PMI-SVD[4]	0.74	0.74
Res95[36]	0.74	0.81
ESA[14]	0.75	0.49
WUP[43]	0.78	0.80
LCH[22]	0.79	0.84
HSO[16]	0.79	0.73
Rad89[34]	0.79	0.79
Taieb [39]	0.80	0.80
BDOS[8]	0.81	–
ZG-07[46]	0.82	0.49
Agirre et al.[1]	0.83	–
Lin[23]	0.834	–
ZMG-08[47]	0.84	–
Hughes and Ramage[17]	0.84	–
Word2vec[28]	0.84	0.83
MP (this approach)	0.87	0.79
coswJ&C [21]	0.876	–

of the time. BDOS and Word2Vec overestimate the semantic similarity of far concepts consistently. This happens because the general knowledge sources like WordNet or corpora always find a path between two concepts in the example of BDOS.

Fig. 6 shows the error of the analysed approaches. Here we can identify the concepts where the approaches deliver good/bad results. For example, it can be noticed that the error of MP is greater in the mid-range of the semantic similarity and that the reference measures get worse with a rising semantic distance. The ELKB approach has less error for distance word pairs as it returns zero if no distance is found. Thus the missing error for semantically distant word pairs here is due to failure of the approach to handle distant words. BDOS on the other hand has almost no error for close concepts. Here short paths between two concepts can be found in WordNet. But the further the distance of the words, the less accurate BDOS becomes.

4.3 Discussion and Future Work

We can see that the thesaurus based approach (ELKB) estimates well when closely related concepts are the problem, whereas the semantic similarity gets smaller the metric becomes less accurate and unable to connect two concepts. In Fig. 6 we can see that ELKB's error increases with the decrease of semantic

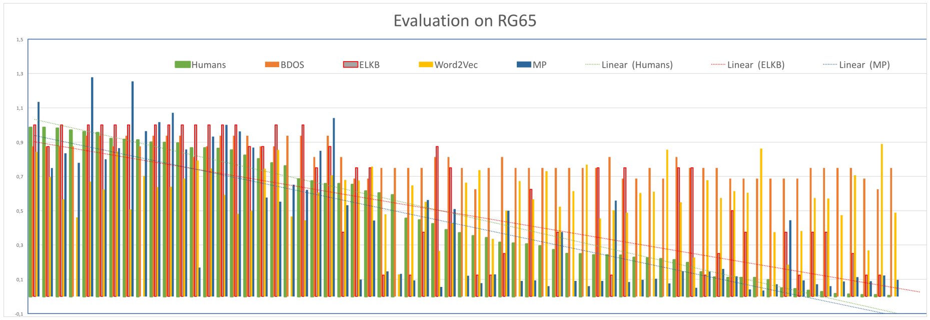

Fig. 5. Evaluation result showing the RG65 dataset.

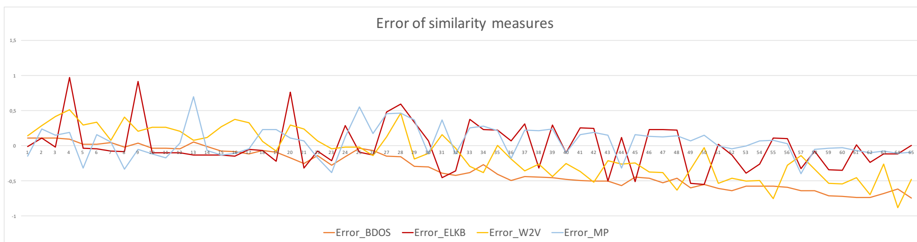

Fig. 6. Evaluation error of the similarity measures tested on the RG65 dataset

similarity of the concepts. The WordNet-Path-length approach (BDOS) on the other hand finds nearly the same similarity for close as for distend concepts.

The use of the RG65 [38] dataset seams insufficient since only nouns are compared. More complex datasets exist, which could be used in the future for comparison. Using part-of-speech independent datasets will worsen thesauri approaches like ELKB since mostly nouns are formalised in thesauri.

A preliminary analysis with the Stanford Rare Word Similarity dataset [26] of 2034 words has yield the result of Spearman correlation of 0.17 and a Pearson correlation of 0.19, which are subject to improvement now. Furthermore, the extension to have a WSD algorithm which uses context can be created through the following steps: The contextual words are decomposed, the results are merged into the graph and markers are passed to them. Then the word sense is selected with the most semantic similarity by identifying the nodes that received the most activation from multiple origins. The parameters for such an algorithm are subject to future work.

In the progress of selecting actions in unknown domains, an agent needs to handle new concepts autonomously. The algorithm proposed here can be seen as a foundation for the integration of new concepts into the beliefs of an agent. Thereby, new concepts and their relations are added to an existing and ever-growing semantic graph.

Future work will include more complex datasets and extending the matching from the worst case to a less general but more informative case where addition information is given. The context dependent decomposition of the concept that is unknown to the agent, and with that a word sense disambiguation for the context of the agent will be a future challenge.

References

1. Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Paşca, M., Soroa, A.: A study on similarity and relatedness using distributional and WordNet-based approaches. In: NAACL '09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (2009)
2. Agirre, E., Edmonds, P.G.: Word sense disambiguation: Algorithms and applications. Springer Science & Business Media (2007)
3. Bar-Hillel, Y.: The Present Status of Automatic Translation of Languages. In: Advances in Computers Volume 1 (1960)
4. Baroni, M., Dinu, G., Kruszewski, G.: 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) (2014)
5. Berthold, M.R., Brandes, U., Kötter, T., Mader, M., 0002, U.N., Thiel, K.: Pure spreading activation is pointless. CIKM (2009)
6. Bruni, E., Tran, N.K., Baroni, M.: Multimodal Distributional Semantics. J Artif Intell Res(JAIR) (2014)
7. Charniak, E.: A Neat Theory of Marker Passing. AAAI (1986)
8. Chen, D., Jianzhuo, Y., Liying, F., Bin, S.: Measure semantic distance in wordnet based on directed graph search. 2009 International Conference on E-learning, E-Business, Enterprise Information Systems, and E-Government, EEEE 2009 (2009)
9. Crestani, F.: Application of Spreading Activation Techniques in Information Retrieval. Artificial Intelligence Review (1997)
10. Erl, T.: SOA Principles of Service Design. Prentice Hall (2007)
11. Fähndrich, J., Ahrndt, S., Albayrak, S.: Are There Semantic Primes in Formal Languages? DCAI 290(Chapter 46), 397–405 (2014)
12. Fähndrich, J., Ahrndt, S., Albayrak, S.: Formal Language Decomposition into Semantic Primes. ADCAIJ: ADVANCES IN DISTRIBUTED COMPUTING AND ARTIFICIAL INTELLIGENCE JOURNAL 3(8), 56 (Oct 2014)
13. Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppin, E.: Placing search in context: the concept revisited. ACM Trans. Inf. Syst. () (2002)
14. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. IJCAI (2007)
15. Goddard, C., Wierzbicka, A.: Semantic and Lexical Universals. Theory and Empirical Findings, John Benjamins Publishing (Jan 1994)
16. Hirst, G., St-Onge, D.: Lexical chains as representations of context for the detection and correction of malapropisms (1998)
17. Hughes, T., Ramage, D.: Lexical Semantic Relatedness with Random Graph Walks. EMNLP-CoNLL (2007)
18. Ide, N., Veronis, J.: Introduction to the special issue on word sense disambiguation: the state of the art. Computational Linguistics 24(1), 2–40 (Mar 1998)

19. Jarmasz, M., Szpakowicz, S.: Roget ' s Thesaurus and Semantic Similarity Roget ' s Thesaurus Relations as a Measure of Semantic Distance. Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2003) pp. 212–219 (2003)
20. Krizhanovsky, A.A., Lin, F.: Related terms search based on WordNet / Wiktionary and its application in Ontology Matching. arXiv.org (2009)
21. Lastra-Díaz, J.J., García-Serrano, A.: A novel family of IC-based similarity measures with a detailed experimental survey on WordNet. Engineering Applications of Artificial Intelligence (2015)
22. Leacock, C., Miller, G.A., Chodorow, M.: Using corpus statistics and WordNet relations for sense identification. Computational Linguistics (1998)
23. Lin, D.: An information-theoretic definition of similarity. ICML (1998)
24. Lin, D., Zhao, S., Qin, L., Zhou, M.: Identifying synonyms among distributionally similar words. IJCAI International Joint Conference on Artificial Intelligence (4), 1492–1493 (2003)
25. Luetzenberger, M., Kuester, T., Masuch, N., Fähndrich, J.: Multi-Agent System in Practice – When Research Meets Reality. In: Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), Singapore (2016)
26. Luong, M.T., Socher, R., Manning, C.D.: Better word representations with recursive neural networks for morphology. CoNLL-2013 (2013)
27. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and Knowledge-based Measures of Text Semantic Similarity. AAAI (2006)
28. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. arXiv.org (2013)
29. Miller, G.A., Charles, W.G.: Contextual correlates of semantic similarity. Language and Cognitive Processes (2007)
30. Minsky, M.L.: Logical versus analogical or symbolic versus connectionist or neat versus scruffy. AI Magazine (1991)
31. Navigli, R.: Word sense disambiguation: A survey. Computing Surveys (CSUR (2009)
32. Patwardhan, S., Pedersen, T.: Using wordnet-based context vectors to estimate the semantic relatedness of concepts. In: Proceedings of the EACL 2006 Workshop Making Sense of Sense-Bringing Computational Linguistics and Psycholinguistics Together. vol. 1501, pp. 1–8. Citeseer (2006)
33. Pilehvar, M.T., Navigli, R.: From senses to texts: An all-in-one graph-based approach for measuring semantic similarity. Artificial Intelligence 228, 95–128 (Nov 2015)
34. Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and application of a metric on semantic nets. Systems, Man and Cybernetics, IEEE Transactions on 19(1), 17–30 (1989)
35. Radinsky, K., Agichtein, E., Gabrilovich, E., Markovitch, S.: A word at a time: computing word relatedness using temporal semantic analysis. computing word relatedness using temporal semantic analysis, ACM, New York, New York, USA (Mar 2011)
36. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. CoRR abs/cmp-lg/9511007 (1995), http://arxiv.org/abs/cmp-lg/9511007
37. Riemer, N.: The Routledge Handbook of Semantics. Routledge (2015)
38. Rubenstein, H., Goodenough, J.B.: Contextual correlates of synonymy. Communications of the ACM 8(10), 627–633 (Oct 1965)

39. Taieb, M.A.H., Ben Aouicha, M., Ben Hamadou, A.: A new semantic relatedness measurement using WordNet features. Knowledge and Information Systems (2014)
40. Thiel, K., Berthold, M.R.: Node Similarities from Spreading Activation. Bisociative Knowledge Discovery (2012)
41. Wierzbicka, A.: Semantics: Primes and Universals. Oxford University Press, USA (1996)
42. Wilbur, W.J., Sirotkin, K.: The automatic identification of stop words. Journal of information science (1991)
43. Wu, Z., Palmer, M.: Verbs Semantics and Lexical Selection. In: Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics (1994)
44. Yampolskiy, R.V.: AI-Complete, AI-Hard, or AI-Easy - Classification of Problems in AI. MAICS pp. 94–101 (2012)
45. Yang, D., Powers, D.M.: Measuring semantic similarity in the taxonomy of wordnet. In: Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38. pp. 315–322. Australian Computer Society, Inc. (2005)
46. Zesch, T., Gurevych, I.: Analysis of the Wikipedia category graph for NLP applications. In: Proceedings of the TextGraphs-2 ... (2007)
47. Zesch, T., Müller, C., Gurevych, I.: Using Wiktionary for Computing Semantic Relatedness. AAAI (2008)
48. Zesch, T., Gurevych, I.: Wisdom of crowds versus wisdom of linguists - measuring the semantic relatedness of words. Natural Language Engineering 16(1), 25–59 (2010)
49. ZHANG, Z., GENTILE, A.L., Ciravegna, F.: Recent advances in methods of lexical semantic relatedness – a survey. Tech. rep. (2012)