Calculating the Similarity of Short Verb Phrases

Günther Fliedl
Alpen Adria Universitaet Klagenfurt, Department of Applied Computer Science
Universitaetsstraße 65-67, A-9020 Klagenfurt
Tel: +43 (463) 2700 3733 – Fax: +43 (463) 2700 993733 – E-mail: Fliedl@ifit.uni-klu.ac.at

Christian Winkler
Alpen Adria Universität Klagenfurt, Department of Linguistics and Computational Linguistics
Universitaetsstraße 65-67, A-9020 Klagenfurt
Tel: +43 (463) 2700 2814 – Fax: +43 (463) 2700 992814 – E-mail: Christian.Winkler@aau.at

Abstract: Over the last decade, Similarity Calculation (SC) has become a topic of major interest in the realm of Natural Language Processing (NLP), including practical domains like Question Answering (QA) and language understanding, which is of interest amongst others for controling intelligent robots. In the field of Word Sense Disambiguation (WSD), a series of different algorithms have been developed for measuring similarity, Lesk being certainly among the most prominent. As opposed to many applications featuring similarity calculation of nouns, in this paper we rise the question whether the calculation of verbs and short verb phrases can be done in a similar way. The result is, that measuring the similarity of verbs and short verbal phrases has to be tackled employing a completely different strategy. In fact, best results can be obtained with Lesk as well, provided that filtering the input of rules and the number of processing steps are well differentiated with respect to the rather simple calculation of noun similarities.

Key-words: Semantic Similarity and Distance, Similarity Calculation, Short Verb Phrases, Robot Control Systems

1. INTRODUCTION

Determining the *degree of semantic similarity*, or *relatedness*, between two words is an important task in Natural Language Processing (NLP). Similarity measures are used in such applications as word sense disambiguation (WSD), determining discourse structure, text summarization and annotation, information extraction and retrieval, automatic indexing, lexical selection, and automatic correction of word errors in text. Human beings have an innate ability to tell if one word is more similar to a given word than another, e.g. most would agree that the musical senses of *guitar* and *string* are related while *guitar* and *beer* are not.

Budansitsky and Hirst [3] compared different measures of similarity or semantic distance in WordNet: Hirst-St-Onge *hso* [6], Leacock-Chodorow *lch* [10], Resnik *res* [27], Jiang-Conrath *jcn* [7] and Lin [12], examining their performance in a real-word spelling correction system, specifically, malapropism detection. In their corrector, words are disambiguated where possible by accepting senses that are semantically related to possible senses of nearby words. Patwardhan et al. *pat* [21] generalizes the Adapted Lesk Algorithm of Banerjee and Pedersen [2] to a method of word sense disambiguation based on semantic relatedness, and they evaluate a variety of measures of semantic relatedness. Adapted Lesk gloss overlaps are based on the definitions found in WordNet, while the measure of Jiang–Conrath is based on the concept hierarchy of WordNet and corpus statistics. Sinha and Mihalcea [31] describe an unsupervised graph-based method for word sense disambiguation, and present comparative evaluations using several measures of word semantic similarity. They propose a combination of similarity measures given by a graph where they use the similarity metric Jiang-Conrath to draw similarity values between nouns and the similarity metric to draw similarity values between verbs. All the other edges in the graph, including links between adjectives and adverbs, or links across different parts-of-speech, are drawn using the *lesk* measure. The results indicate that the right combination of similarity metrics can lead to a performance competing with the state-of-the-art in unsupervised word sense disambiguation. See also [33].

As far as verbs and verb phrases are concerned, similarity calculation with the Lesk Algorithm proved to be promising as well, provided that the textual input for the calculation rules is based on verb synonym lists. Related senses can be identified by finding overlapping words in their definitions, in that words that are related will often be defined using the same words, and in fact may refer to each other in their definitions.

The practical benefit of this method lies amongst others in the field of machine control. The inventory of rules permits the grouping of orders (which are primarily expressed by verb-noun combinations) with the help of similarity calculation, thus generating pragmatically relevant order classes of robot commands.

2. WORDNET

The semantic relation of words is represented by organizing words into sets of synonyms (synsets) that lexically express concepts. Synonyms are words that have common meaning, e.g. *sofa*, *couch* and *lounge* mean *an upholstered seat for more than one person*. Polysemous words bear several meanings, e.g. *paper* has 7 meanings as a noun and 2 meanings as a verb. Thus, *paper* is a 9-fold polysemous word. Among all nouns defined in WordNet, approximately 41% are polysemous, but it is verbs which show the highest overall polysemy count, namely 3.57 excluding monosemous words [26].

The felicitous meaning of highly polysemous words and synonyms can be determined only within its context. Therefore, a common way of dealing with polysemous words and synonyms consists in limitating them to a certain domain of discourse [16]. For example, in the context of a conference, a paper means *a scholarly article describing the results of observations or stating hypotheses*, but in context of a manufacture it would rather refer to *a material made of cellulose pulp derived mainly from wood or rags or certain grasses*. All nouns are inherited from *entity,* which subsumes the more specific concepts – *abstraction*, *physical entity* and *thing*. To distinguish the senses of a specific concept, the *word#pos#sense* notation is used, where *pos* is a part of speech (n – noun, v – verb, adj – adjective, adv - adverb) and *sense* – number of defined in taxonomy meanings. Indirect descendants (like *abstract entity* and *social group*) are connected with a dash line. It might be interesting to point out that both the sub-trees of *social group* and *construction* comprise *room* and *house* as successors. *Room* in the *social group* hierarchy means *the people who are present in a room*, but in the *construction* sub-trees it is defined as *an area within a building enclosed by walls and floor and ceiling*. The meanings of *house* are *the audience gathered together in a theatre or cinema* and *a dwelling that serves as living quarters for one or more families* respectively.

3. WORD SENSE DISAMBIGUATION

Most nouns in natural languages are polysemous, that means they have multiple senses. For example, *check* may refer to a bill in a restaurant, an act of inspecting, or verifying. Even during a conversation, the lack of context can lead to difficulties when identifying the meaning of ambiguous words (e.g., *ask for a check*). With the help of the surrounding context, humans immediately determine the implied meaning (for example, *your paper needs additional check*). As for computers, however, Word Sense Disambiguation (WSD), i.e. automatically assigning the appropriate sense to a polysemous word, is a non-trivial task.

WSD processing can be performed by assigning the similarity or equality of the elements manually by domain experts, or (semi)automatically by making use of a domain-ontology. Knowledge is a fundamental component of WSD, relying on different knowledge sources that provide data to associate senses with words. Navigli [18] gives an overview of the available resources according to which unstructured sources comprise corpora (collections of texts used for a learning language model), collocation resources (register the tendency for words to occur regularly with others), and others.

Accordingly, structured sources provide information about relationships between words, like synonymy. Structured knowledge sources are:

- Machine-readable dictionaries, or dictionaries in electronic format
- Thesauri (provide information about relationships between words)
- Ontologies (specifications of conceptualizations of specific domains of interest)

General-purpose lexicons give domain-independent definitions. As for the English language, WordNet is a free lexicon, providing definitions for nouns, verbs, adjectives and adverbs, allowing word sense extraction as well. Since it encodes a rich semantic network of concepts, WordNet is often considered one step beyond machine-readable dictionaries.

4. MEASURING WORD SIMILARITY

The general process of measuring the similarity is quite simple: having two concepts as an input, an algorithm is applied to these concepts based on WordNet database, in order to obtain a specific similarity score. After that, a decision has to be made with respect to the examined concepts. Typically, if the score is higher than the predefined threshold, we can assume that the concepts are similar. In some cases, the confirmation of the domain expert is required. Vöhringer and Fliedl [34] discuss the following basic similarity measures:

- Path length evaluation (Leacock-Chodorow, Wu-Palmer)
- Information content and least common subsumer evaluation (Resnik, Jiang and Conrath, Lin)
- Semantic relatedness evaluation (Hirst and St-Onge, Lesk)

Some approaches combine different measures or expand them. Others, as e.g. the vector measure, are discussed in Pedersen [23]. In the current project[1], similarly to Metzler et al. [15] and Jiang an Conrath [7], eight basic measures are taken for assessment, since different results are given in the papers. For measuring similarity the Perl package WordNet::Similarity[2] [26] is freely available. For comparison we used open Java API WS4J[3].

4.1. Noun comparison

For assessing measures three pairs from home environment domain were compared: *table – desk*, *table – chair*, *table – lamp*.

In Table 1, number scores of basic measures are given. The ranges of measures in WordNet implementation are taken from Miller [16].

Table 1. Noun comparison

Measure with range	Concept pair		
	table – desk	*table – chair*	*table – lamp*
wp [0; 1]	0.9524	0.8571	0.7000
lch [0; 3.6889)	2.9957	2.3026	1.7430
pat [0,1]	0.5000	**0.2500**	0.1429
res [0; ∞)	7.4091	6.1860	3.4451
jcn [0; 1]	0.9102	**0.3411**	0.1138
lin [0; 1]	0.9310	0.8084	0.4395
hso [0; 16)	4	***5***	2
lesk [0; ∞)	738	**274**	167

Almost all measures identified the *table – desk* pair as the most similar. Only *hso* has the maximum similarity (=***5***) for *table – chair* pair. Although the *table – chair* pair is related to the *furniture* sub-tree, *pat*, *jc* and *lesk* have surprisingly low scores (**bold**) with respecpt to *table – desk*. Consistently, all measures reflect that *table – lamp* pair is less related than *table – chair*.

Since the *lesk* algorithm depends on the definitions in taxonomy, the upper bound of this measure is not strictly defined. For the examples considered the same nouns are taken, given the fact that the focus of this work lies on verbs and the comparison of verb phrases. It is necessary to choose empirically the threshold value *x,* so that all nouns with a score less than *x* will be considered as dissimilar. In such cases is not necessary to perform verb comparison.

To compare different nouns within one domain, the maximum value among different nouns should be taken for normalization. For example, when comparing *coffee – water* and *coffee – coffee* with *lesk*, the score in the first case will be lower, because the same definitions lead to a high score in the second compared pair. For the analysis of phrase and noun similarity, equal concepts must be normalized (=1) and filtered out. Likewise, normalization concerning the most similar concepts in the considered domain should be carried out.

Lesk measure initially proposed that the relatedness of two words is proportional to the extent of overlaps of their dictionary definitions [11]. This implies that the *lesk* metric is not dependant on the taxonomy structure. Instead

[1] All testing and evaluating has been performed as part of the master thesis by *Natalia Bilogrud*, who has also designed the tables and figures contained in the paper, and written the Java program for generating the scores. The authers wish to express their gratitude.

[2] http://wn-similarity.sourceforge.net/

[3] http://code.google.com/p/ws4j/

of this, the definitions, or glosses of a word are used. Every word in a phrase is compared to the glosses of every other word in the phrase. The measure is higher when compared words have the largest number of words in their definitions in common. For example, in *a journey begins with a single step*, the algorithm compares the glosses of *journey* to all the glosses of *begins, single, step* and so on. Then, the algorithm begins anew for each word, skipping the senses previously assigned [2].

During calculation *lesk*, such types of glosses for a word *a* are considered:

- example – example of usage;
- gloss – word sense definition;
- hype gloss – "a is a kind of …";
- hypo gloss – "… is a kind of *a*";
- holo gloss – "a is a part of …";
- mero gloss – "… is a part of *a*".

When overlaps are determined, the overlap score is calculated:

$$overlapScore = nOccurences * overlapLength,$$

where *nOccurences* – number of occurences, *overlapLength* – amount of words in common. Then the sum of overlap scores is calculated:

$$glossPermutationScore = \sum overlapScore$$

The total *lesk* score is calculated as a sum of gloss permutation scores:

$$lesk = \sum glossPermutationScore$$

At first sight, *Lesk* metric seems simple and intuitive, yet it is dependent on the number and the size of the glosses available in the taxonomy. As a result, *table* and *desk* have high scores; *table* is more similar to *chair* than to *lamp* – which is correct within the furniture domain. If we compare *move* (*cause to move or shift into a new position or place, both in a concrete and in an abstract sense*) with *stir* (*move an implement through*) and *turn on* (*cause to operate by flipping a switch*) the results obtained in the database are also promising.

With respect to 3 meanings (out of 25) of the word *check* the glosses are as follows:

- S: (n) check, chit, tab (*the bill in a restaurant*)
- S: (n) confirmation, verification, check, substantiation (*additional proof that something that was believed (some fact or hypothesis or theory) is correct*)
- S: (v) see, check, insure, see to it, ensure, control, ascertain, assure (*be careful or certain to do something; make certain of something*)

The above example shows that the word *check* occurs in two noun (n) synsets *{check, chit, tab}*, *{confirmation, verification, check, substantiation}* and one verb (v) synset *{see, check, insure, see to it, ensure, control, ascertain, assure}*. Each concept has a short gloss (for example, *the bill in a restaurant*), and most of them have an example usage (e.g. example, *he asked the waiter for the check*). Since verbs behave completely different than nouns, they have to be analyzed and calculated in a different way.

4.2 Verb comparison

For analysing verbs concepts, five pairs of verbs (from Fig. 3) are compared:

Ex1 – (move#v#1; move#v#1)
Ex2 – (move#v#2; stir#v#1)
Ex3 – (move#v#2; turn_on#v#1)
Ex4 – (move#v#1; sleep#v#1)
Ex5 – (move#v#1; hope#v#1)

Table 2. Verb comparison

Measure with range	Ex1	Ex2	Ex3	Ex4	Ex5
wp [0; 1]	1.0000	0.8	0.5714	0.2857	0.3333
lch [0; 3.6889)	3.3322	2.6391	1.9459	1.5404	1.7228
pat [0,1]	1.0000	0.5	0.2500	0.1667	0.2000
res [0; ∞)	3.3962	3.2589	3.2589	0.0000	0.0000
jcn [0; 1]	9593499.5	0.1630	0.1859	0.0940	0.0934
lin [0; 1]	1.0000	0.5152	0.5478	0.0000	0.0000
hso [0; 16)	16	4	5	0	0
lesk [0; ∞)	7533	491	40	50	4

For equal concepts, as expected, the highest values were obtained. Although verbs like *move* and *sleep*, *move* and *hope* apparently are completely dissimilar, only *res* and *lin* reflect this. If we consider the definitions of *move* and *stir* (*cause to move or shift into a new position or place, both in a concrete and in an abstract sense* and *move an implement through*), the results given by *wp*, *lc*, *res* and *lesk* are realistic. As for the definitions of the phrasal verb *turn on* (*cause to operate by flipping a switch*), it is difficult to compare with *move* without a pre-defined context.

Because of the ambiguous structure of verb taxonomy, *lesk* is the most appropriate algorithm for comparing verbs. This measure is not dependent on the WordNet structure but based on glosses defined in WordNet. Another advantage of *lesk* is its optimization potential regarding internal and external factors. Possible optimization strategies include partial filtering of stop words, word reduction via stemming, normalization based on gloss length, improvement of glossary quality and quantity via completion and substitution of certain keywords, restructuring of taxonomy and others.

A configuration file for the *lesk* measure allows to adjust trace (turn off/on), cache (turn off/on caching), maxCacheSize (limit the cache size to 1000 pairs of query words), stem (turn stemming on), relation (specifies the path to a lesk relation file to be used), stop (specifies the path to a list of stop words that should be ignored for the gloss overlaps), normalize (disable/allow a normalisation).

A stop list is a list of words that are excluded from some language-processing task, because they are viewed as non-informative or potentially misleading and are often called function words. For evaluating *lesk* measure concepts are compared with and without stop words. Table 3 illustrates the results of comparison the pairs without pre-defined definition in WordNet and part of speech.

Table 3. Comparison (max, default) concepts

Concept pair	With stop words	Without stop words
table – desk	(#n#2 – #n#1) 738	(#n#2 – #n#1) 1.275
table – chair	(#n#2 – #n#1) 274	(#n#2 – #n#1) 0.247
table – lamp	(#n#2 – #n#2) 266	(#n#2 – #n#2) 1.121
move – move	(#v#1 – #v#1) 7533	(#n#2 – #n#2) 3.672
move – stir	(#v#2 – #v#1) 491	(#v#3 – #v#2) 1.605
move – turn_on	(#v#13 – #v#4) 172	(#v#13 – #v#4) 1.056
move – sleep	(#n#3 – #n#1) 56	(#v#6 – #v#1) 0.333
move – hope	(#v#2 – #n#4) 35	(#v#13 – #n#4) 0.167

By default *lesk* compares the concepts with the utmost amount of definitions. In the table above, the corresponding positions of compared concepts are given in the brackets before the output scores. Depending on the calculation including or excluding stop words, even the POS of compared words can differ (e.g. *move – move*, *move – sleep*

in table 3). The pairs *table – chair* and *table – lamp* are similar when compared including stop words, but totally dissimilar when excluding stop words.

Apparently, the correct identification of POS for single words and for short phrases is not possible without context. Since in WordNet, however, the most frequent concepts are defined in the first position of the glosses, such a comparison will not consider any context. The results obtained are shown in Table 4.

Table 4. Comparison the most frequent concepts (#n#1, #v#1)

Concept pair	With stop words	Without stop words
table – desk	66	0.018
table – chair	78	0.009
table – lamp	66	0.0002
move – move	7533	1.576
move – stir	49	0.201
move – turn_on	27	0.001
move – sleep	50	0.008
move – hope	4	0

Thus, this approach implies the risk that wrong data might be analyzed. In the given example, the presupposition is that the definition of *table* concerns the furniture domain. Yet according to WordNet the first definition of *table* is *a set of data arranged in rows and columns*, so that the outcome is quite useless for the case considered here.

Table 5. Comparison pre-defined concepts

Concept pair	With stop words	Without stop words
table#n#2 – desk#n#1	738	1.275
table#n#2 – chair#n#1	274	0.247
table#n#2 – lamp#n#1	165	0.004
move#v#1 – move#v#1	7533	1.576
move#v#2 – stir#v#1	491	1.268
move#v#2 – turn_on#v#1	40	0.050
move#v#1 – sleep#v#1	32	0.008
move#v#1 – hope#v#1	6	0

To avoid such deficiencies, concepts for comparison must be identified according to the respective number of definitions. Table 5 shows the evaluation results performed with pre-defined concepts.

Verbs are arguably the most sophisticated lexical and syntactic category of a language. Without doubt, verb meanings are even more variegated than those of nouns; which is confirmed by the highest polysemous count of verbs. The most frequent verbs (like *be*, *have*, *make*, *go*, *take*) heavily depend on the nouns in the context of which they occur. Currently, WordNet contains over 25.000 verb forms, and 13.767 verb synsets. Phrasal verbs like *turn on*, *fall back* are also included. Since WordNet does not perform any semantic decomposition of verbs but focuses on relational analysis instead, these verbs cannot be organized hierarchically in a similar way as nouns are defined: we do not say *to walk is a to move*). Therefore, verbs in WordNet are grouped in 15 files based on semantic criteria: verbs of bodily care and functions, change, cognition, communication, competition, consumption, contact, creation, emotion, motion, perception, possession, social interaction, and weather verbs. But even in such a taxonomy not all verbs can be grouped into a single unique beginner. Motion verbs, for instance have two top nodes: *{move, make a movement}*, and *{move, travel}*.

Consequently, verb hierarchy tends to have a more shallow structure than that of nouns, see Figure 1 for part of the verb hierarchy. The dotted line connects polysemous verbs.

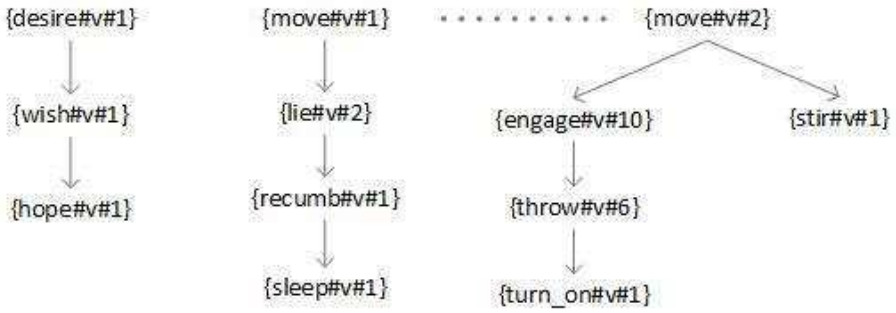

Figure 1. Fragment of the WordNet verb taxonomy

The troponymy relation between two verbs v_1 and v_2 could be paraphrased as v_1 *is to* v_2 *in some particular manner,* e.g. *limp – walk*. *To limp* is also *to walk* in a certain manner. Unlike hypernym relations, troponymy relations are temporally coextensive. All semantic relations among verbs interact with entailment. For instance, in both pairs *drive – ride* and *snore – sleep* the first activity entails the second. Adjectives, adverbs and some verbs are arranged into antonym (*is-opposite-of*) relations, specifically, words can be direct (small *is opposite of* large) and indirect (flying *is opposite of* unhurried) antonyms. Further relations defined in WordNet are described in Miller [16], [26].

5. SHORT VERB PHRASES COMPARISON

The following schema visualizes the proposed comparison. It mainly reflects the diversification between noun and verb comparison during the workflow.

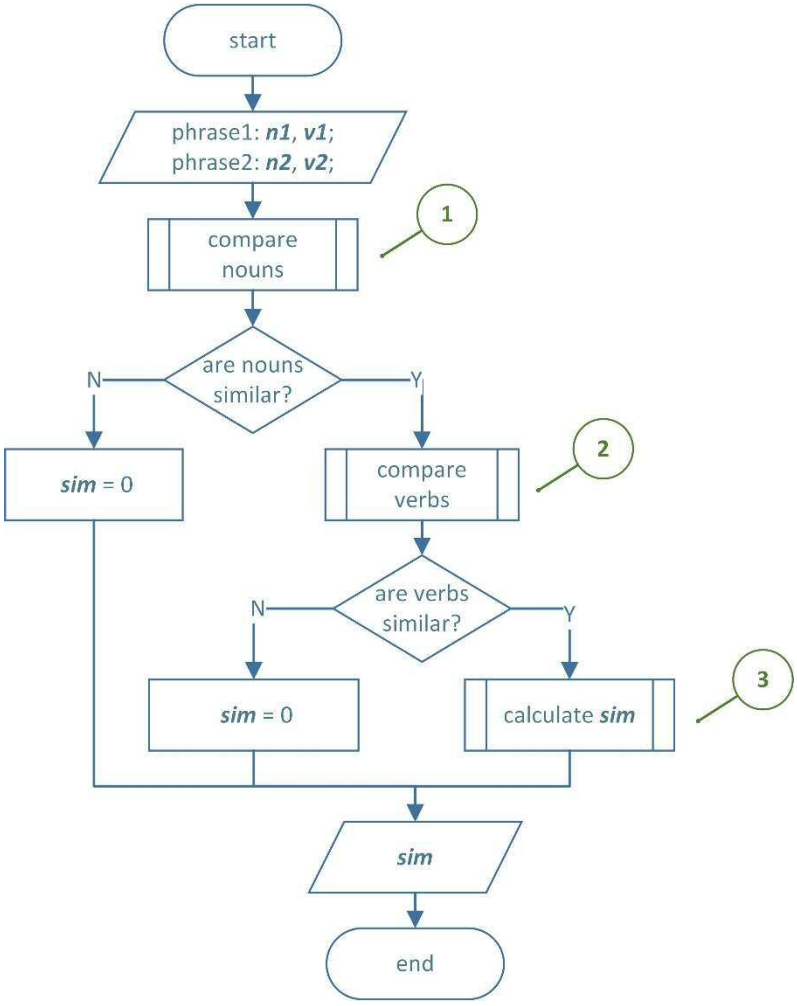

Figure 2. Block schema for verb phrases comparison

For the comparison of verb phrases, 10 triples are analyzed, pre-supposing different semantic distance. Thus, *take a cup* will be compared with *bring a cup* and *clean a cup.* All phrases are taken from the kitchen domain.

5.1 Determining the identity of involved nouns

Since nouns in a phrase represent entities, and verbs express relations, first nouns should be matched.

Table 6. Compared phrases

1	take a cup	bring a cup	clean a cup
2	turn on a kettle	switch on a kettle	wash a kettle
3	drink a coffee	taste a coffee	add coffee
4	wipe a table	clean a table	cover a table
5	bake a cake	cook a cake	eat a cake
6	push a button	press a button	break a button
7	prepare a breakfast	make a breakfast	wait a breakfast
8	boil water	heat water	pour water
9	pour milk	add milk	take out milk
10	take a spoon	get a spoon	wash a spoon

Then, if nouns are semantically similar or equivalent, verbs must be compared; otherwise, the similarity of phrases, as defined in this workflow, will be equal 0.

5.2. Similarity scores for verbs

In the next step, preliminary results are obtained by calculating the similarity scores for verbs with *lesk*:

Table 7. Verb comparison with *lesk*

	verb 1	verb 2	score		verb 1	verb 2	score
1-a	take	bring	0.684	1-b	take	clean	0.004
2-a	turn on	switch on	2.503	2-b	turn on	wash	0
3-a	drink	taste	0	3-b	drink	add	0
4-a	wipe	clean	0.011	4-b	wipe	cover	0.0002
5-a	bake	cook	1.157	5-b	bake	eat	0.007
6-a	press	push	0.314	6-b	press	break	0.113
7-a	prepare	make	2.429	7-b	prepare	wait	0.0008
8-a	boil	heat	0.189	8-b	boil	pour	0.252
9-a	pour	add	0.001	9-b	pour	take out	0.032
10-a	take	get	0.238	10-b	take	wash	0.010

As can be deduced from the table above, the pairs in *b* suggest to be much less similar than those in *a*. Apparently, the resulting scores reflect horizontal differences in the respective line (1-10) in a plausible way. Similiarty in the vertical columns however cannot be interpreted without previous normalisation.

5.3. Synonym lists

In order to obtain the most suitable synonym list many items are taken from *The WordReference English Thesaurus* [35]. There are different strategies suitable for extracting synonyms: based on either the co-occurrence with WordNet-synonyms, or on the definition in WordNet, or manually selected. Without doubt, the last one is the most robust, yet it requires a qualified linguist.

Verb pairs should be compared with respect to their cross-synonyms and common synonyms. A verb *v1* has a cross-synonym with *v2* if the list of synonyms of *v1* contains *v2*. Pairs can be one- or two-directional. As a consequence, verbs with two-directional synonyms are more likely to be similar than those with one (Figure 3).

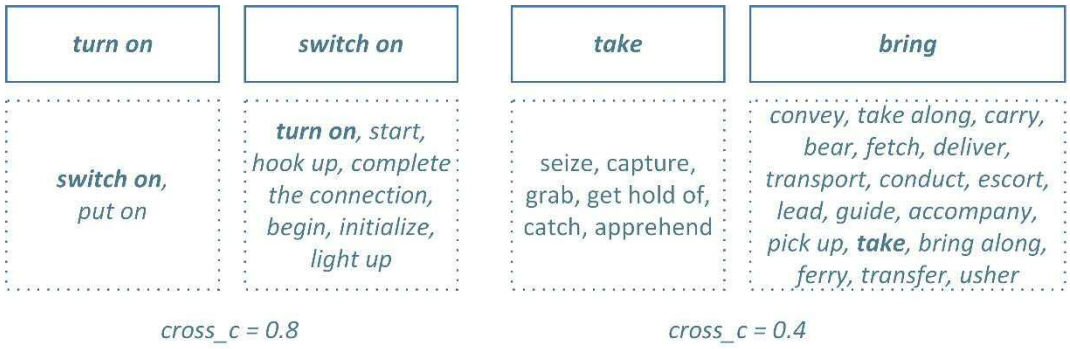

Figure 3. Cross synonyms

The weight of common synonyms should be dependent on the total amount of synonyms for a certain verb (Figure 4). As for verbs without cross-synonyms, it is recommendable to consider a number of common synonyms. For example, *taste* and *drink* have 0 *lesk* score and 3 synonyms in common. In such cases, adding the constant value *x* to the initial verb score, and notifying the domain expert to check these pairs manually, *drink – taste* and *take – get* will lead to their correct identification as similar, which is true for the current example.

Figure 4. Common synonyms

Table 8 shows the list of synonyms for the considered verbs taken from WordReference with highlighted cross-synonyms and common synonyms.

$$k = com_c - 0.01 * (list1_s + list2_s - com) + cross_c$$

(second option $sim_v = lesk + cross_c + x * amount_{com}$)

Table 8. Synonyms from WordRef

№	compared phrases	amount for v1- v2 (common)	cross-syn	
			v1	v2
1-a	take cup - bring cup	6 – 18 (0)		+
1-b	take cup - clean cup	6 – 79 (0)		
2-a	turn on kettle - switch on kettle	2 – 7 (0)	+	+
2-b	turn on kettle - wash kettle	2 – 23 (0)		
3-a	drink coffee - taste coffee	**27 – 15 (3)**		
3-b	drink coffee - add coffee	27 – 15 (3)		

4-a	wipe table - clean table	13 – 79 (5)	+	+
4-b	wipe table - cover table	13 – 18 (0)		
5-a	bake cake - cook cake	8 – 68 (5)	+	+
5-b	bake cake - eat cake	8 – 92 (0)		
6-a	press button - push button	98 – 11 (4)	+	
6-b	press button - break button	99 – 37 (1)		
7-a	prepare breakfast - make breakfast	30 – 5 (1)		+
7-b	prepare breakfast - wait breakfast	30 – 27 (0)		
8-a	boil water - heat water	22 – 6 (0)	+	
8-b	boil water - pour water	22 – 4 (0)		
9-a	pour milk - add milk	4 – 15 (0)		
9-b	pour milk - take out milk	4 – 5 (0)		
10-a	take spoon - get spoon	**6 – 7 (1)**		
10-b	take spoon - wash spoon	5 – 23 (0)		

5.4 Overall score calculation

For optimal comparison, the verb (and noun) score should be normalized relatively the current domain. In the current example, the maximum value for nouns is 2.503, and 4.056 for verbs. The final score is an arithmetic average of normalized noun and verb scores[4] (Table 9).

$$sim_{final} = \frac{sim_n + sim_v}{2}$$

Table 9. Final scores

№	compared phrases	final score	№	compared phrases	final score
1-a	take cup bring cup	0.7166	1-b	take cup clean cup	0.0758
2-a	turn on kettle switch on kettle	1.0	2-b	turn on kettle wash kettle	0.375
3-a	drink coffee taste coffee	0.705	3-b	drink coffee add coffee	0.29
4-a	wipe table clean table	0.9422	4-b	wipe table cover table	0.345
5-a	bake cake cook cake	1.0	5-b	bake cake eat cake	0.2014
6-a	press button push button	0.7127	6-b	press button break button	0.0976
7-a	prepare breakfast make breakfast	1.0	7-b	prepare breakfast wait breakfast	0.2152
8-a	boil water heat water	0.5978	8-b	boil water pour water	0.4203
9-a	pour milk add milk	0.4052	9-b	pour milk take out milk	0.4614
10-a	take spoon get spoon	0.7375	10-b	take spoon wash spoon	0.357

[4] *Lesk* scores were extracted with a WordNet Perl-script adopted by Jürgen Vöhringer [34]. Currently, the following input values are required: *pairId, phrase1, phrase2, verb1, verb2, noun1, noun2, lesk_verb, lesk_noun, synonym_list1, synonym_list2*, delivering the output *pairId* and final *score*. Final scores were generated with the help of a java program written by Natalia Bilogrud (see FN1).

In a final step, domain experts have been asked to check the results, and they confirmed the validity of the scores and thus their empirical plausibility. Due to the ambiguity of language, considering especially too high genericity and/or too little specificity of terms, the method proposed in this paper will produce best results if employed in a domain specific environment.

CONCLUSION

As our experiments have shown, similarity calculation with the Lesk Algorithm can be applied perfectly to verbs and verb phrases. In such case, however, it is mandatory that the textual input for the calculation rules consists of a verb synonym list, as opposed to concept definitions, which are used in the context of nouns. The degree of similarity is calculated by the cover ratio of the lists employed, as well as by the occurrence of both verbs in the respective counterpart list. The practical benefit of this method lies amongst others in the field of machine control. The inventory of rules permits the grouping of orders (which are primarily expressed by verb-noun combinations) with the help of similarity calculation, thus generating pragmatically relevant order classes of robot commands. Such an algorithm based system of command generation can easily cover domain relevant sequences of interactions including the synonym guided usage of natural language.

REFERENCES

[1] E. Agirre and Edmonds P., Eds. *Word Sense Disambiguation: Algorithms and applications*. Text, Speech and Language Technology Series, Springer, 2007, Vol. 33, ISBN: 978-1-4020-6870-6.

[2] S. Banerjee and T. Pedersen. An adapted Lesk algorithm for word sense disambiguation using WordNet. In *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 136–145, CICLing 2002, Mexico City, February 2002.

[3] Budanitsky and G. Hirst. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and Other Lexical Resources, Second meeting of the North American Chapter of the Association for Computational Linguistics*, Pittsburgh, June 2001.

[4] S. Cai and Z. Lu, "An Improved Semantic Similarity Measure for Word Pairs," *2010 Int. Conf. e-Education, e-Business, e-Management e-Learning*, pp. 212–216, 2010.

[5] C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[6] G. Hirst and D. St-Onge, Lexical chains as representations of context for the detection and correction of malapropisms. In: C. Fellbaum (Ed.), *WordNet: An electronic lexical database*, Cambridge, MA: The MIT Press, 1998, 305–332.

[7] J. Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics*, Taiwan, 1997.

[8] A. Kilgarriff, "I don't believe in word senses". https://www.sketchengine.co.uk/wp-content/uploads/I_dont_believe_1997.pdf

[9] A. Kilgarriff, "English lexical sample task description," *Proc. Second Int. Work. ...*, pp. 17–20, 2001. https://www.kilgarriff.co.uk/Publications/2001-K-Senseval2EngLexSamp.pdf

[10] Leacock and M. Chodorow. Combining local context and WordNet sense similarity for word sense identification. In *WordNet, An Electronic Lexical Database*. The MIT Press, 1998.

[11] M. Lesk. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a ice cream cone. In *Proceedings of SIGDOC '86*, 1986.

[12] Lin. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, Madison, WI, 1998.

[13] Z. Liu, X. Chen. "A Graph-Based Text Similarity Algorithm," National Conference on Information Technology and Computer Science (CITCS 2012), pp. 614–617, 2012.

[14] J. G. Mersch and R. R. Lang, "Comparison of Algorithmic and Human Assessments of Sentence Similarity" International Joint Conference on Natural Language Processing, pages 1306–1311, Nagoya, Japan, 14-18 October 2013.

[15] D. Metzler, S. Dumais, and C. Meek, "Similarity Measures for Short Segments of Text." http://research.microsoft.com/en-us/um/people/sdumais/ECIR07-MetzlerDumaisMeek-Final.pdf

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

[17] J. Morato, M. Á. Marzal, J. Lloréns, and J. Moreiro, "WordNet Applications," pp. 270–278, 2004.

[18] R. Navigli, "Word sense disambiguation," *ACM Comput. Surv.*, vol. 41, no. 2, pp. 1–69, Feb. 2009.

[19] M. Palmer, C. Fellbaum, S. Cotton, L. Delfs, and H. Dang. English tasks: allwords and verb lexical sample. In *Proceedings of ACL/SIGLEX Senseval-2*, Toulouse, France, 2001.

[20] M. Palmer and M. Light. Introduction to the special issue on semantic tagging. *Natural Language Engineering*, 5(2): i-iv.

[21] S. Patwardhan, S. Banerjee, and T. Pedersen. Using Measures of SemanticRelatedness for Word Sense Disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 241–257, Mexico City, Mexico, February, 2003.
[22] M. T. Pazienza, M. Pennacchiotti, F. M. Zanzotto, and V. Politecnico, "Mixing WordNet , VerbNet and PropBank for studying verb relations." http://www.cs.brandeis.edu/~marc/misc/proceedings/lrec-2006/pdf/379_pdf.pdf
[23] T. Pedersen, "Siddharth Patwardhan," 2003.
[24] T. Pedersen, S. Patwardhan, and J. Michelizzi, "WordNet::Similarity: measuring the relatedness of concepts," pp. 38–41, May 2004.
[25] S. Pradhan, E. Loper, D. Dligach, and M. Palmer. SemEval-2007 task 17: English lexical sample, SRL and all words. In *Proceedings of the 4th international workshop on semantic evaluations (SemEval-2007),* pp. 87–92, Prague, Czech Republic, 2007.
[26] Princeton University, "WordNet. A lexical database for English." [Online]. Available: http://wordnet.princeton.edu/.
[27] P. Resnik, "WordNet and Distributional Analysis : A Class-based Approach to Lexical Discovery," 1992.
[28] P. Resnik. Using information content to evaluate semantic similarity. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Montreal, Canada, 1995.
[29] M. Sahami and T. D. Heilman, "A web-based kernel function for measuring the similarity of short text snippets," *Proc. 15th Int. Conf. World Wide Web - WWW '06*, p. 377, 2006.
[30] Sebti and A. Barfroush. A new word sense similarity measure in wordnet. In *Proceedings of the IEEE International Multiconference on Computer Science and Information Technology*, October, 2008.
[31] R. Sinha and R. Mihalcea. Unsupervised Graph-based Word Sense Disambiguation Using Measures of Word Semantic Similarity, In *Proceedings of the IEEE International Conference on Semantic Computing (ICSC 2007)*, Irvine, CA, September 2007.
[32] B. Snyder and M. Palmer. The English all-words task. In *Proceedings of ACL/SIGLEX Senseval-3*, Barcelona, Spain, July 2004.
[33] S. Torres and A. Gelbukh. ComparingSimilarity Measures for Original WSD Lesk Algorithm. *Advances in Computer Science and Applications. Research in Computing Science 43*, 2009 pp. 155-166
[34] J. Vöhringer and G. Fliedl (2011). *Adapting the Lesk Algorithm for Calculating Term Similarity in the Context of Ontology Engineering*. Proceedings of ISD2010 Conference.
[35] WordReference. http://www.wordreference.com/