A Novel Methodology For Measuring Semantic Similarity Of Words Using Concept Networks

Farooq Ahmad[1] , Mohammad Faisal[2]

[1,2]Department of Computer Application, Integral University, Lucknow, India.

Abstract

The problem of measuring sentence similarity is an essential issue in the natural language processing area. Semantic similarity between words is a crucial task for many applications. The emerging of word embedding encourages calculating similarity between words based on the new semantic word representation. On the other hand, WordNet is widely used to find semantic distance between words. There are many approaches to measuring sentence similarity. The semantic analysis field has a crucial role to play in the research related to the text analytics. The semantic similarity differs as the domain of operation differs. In this study, we propose a state-of-the-art algorithm for measuring the semantic similarity of word pairs using novel combinations of word embeddings, WordNet, and the concept dictionary 4lang. We evaluate our system on the SimLex-999 benchmark data. Our top score of 0.76 is higher than any published system that we are aware of, well beyond the average inter-annotator agreement of 0.67, and close to the 0.78 average correlation between a human rater and the average of all other ratings, suggesting that our system has achieved near- human performance on this benchmark.

Keywords: Natural Language Processing, WordNet, Word Embedding, Semantic Similarity, 4lang

Introduction

We present a hybrid system for measuring the se- mantic similarity of word pairs. The system relies both on standard word embeddings, the WordNet database, and features derived from the 4lang concept dictionary, a set of concept graphs built from entries in monolingual dictionaries of English. 4lang-based features improve the performance of systems using only word embeddings and/or WordNet, our top configurations achieve state-of-the-art results on the SimLex-999 data, which has recently become a popular benchmark of word similarity metrics.

In Section 1 we summarize earlier work on measuring word similarity and review the latest results achieved on the SimLex-999 data. Section 2 describes our experimental setup, Sections 2.1 and 2.2 documents the features obtained using word embeddings and WordNet. In Section 3 we briefly introduce the 4lang resources and the formalism it uses for encoding the meaning of words as directed graphs of concepts, then document our efforts to develop novel 4lang-based similarity features. Besides improving the performance of existing systems for measuring word similarity, the goal of the present project is to examine the potential of 4lang representations in representing non-trivial lexical relationships that are beyond the scope of word embeddings and standard linguistic ontology's.

Section 4 presents our results and provides rough error analysis. Section 5 offers some conclusions and plans for future work. All software presented in this paper is available for download under an MIT license at http://github.com/recski/wordsim.

1 Background

Measuring the semantic similarity of words is a fundamental task in various natural language processing applications. The ability to judge the similarity in meaning of any two linguistic structures reflects on the quality of the representations used. Vector representations (word embeddings) are commonly used as the component encoding (lexical) semantics in virtually all NLP applications. The similarity of word vectors is by far the most common source of information for semantic similarity in state-of-the-art systems, e.g. nearly all top-scoring systems at the 2015 SemEval Task on measuring semantic similarity (Agirre et al., 2015) rely on word embeddings to score sentence pairs (see e.g. (Sultan et al., 2015; Han et al., 2015)).

Hill et al. (2015) proposed the SimLex-999 dataset as a benchmark for word similarity, arguing that pre-existing gold standards measure association, not similarity, of word pairs; e.g. the words cup and coffee receive a high score by annotators in the widely used wordsim353 data (Finkelstein et al., 2002). SimLex has since been used to evaluate various algorithms for measuring word similarity. Hill et al. (2015) reports a Spearman correlation of 0.414 achieved by an embedding trained on Wikipedia using word2vec (Mikolov et al., 2013). Schwartz et al. (2015) achieves a score of 0.56 using a combination of a standard word2vec-based embedding and the SP model, which encodes the co occurrence of words in symmetric patterns such as X and Y or X as well as Y.

Banjade et al. (2015) combined multiple word embeddings with the word similarity algorithm of (Han et al., 2015) used in a top-scoring SemEval system, and simple features derived from WordNet (Miller, 1995) indicating whether word pairs are synonymous or antonymous. Their top sys- tem achieved a correlation of 0.64 on SimLex. The highest score we are aware of is achieved using the Paragram embedding (Wieting et al., 2015), a set of vectors obtained by training pre-existing embeddings on word pairs from the Paraphrase Database (Ganitkevitch et al., 2013). The top correlation of 0.69 is measured when using 300-dimension embedding created from the same GloVe-vectors that have been introduced in this section (trained on 840 billion tokens). Hyper-parameters of this database have been tuned for maximum performance on SimLex, another version tuned for the WS-353 dataset achieves a correlation of 0.667.

2 Setup

Our system is trained on a variety of real-valued and binary features generated using word embeddings, WordNet, and 4lang definition graphs. Each class of features will be presented in detail below. We perform support vector regression (with RBF kernel) over all features using the numpy library, the model is trained on 900 pairs of the SimLex data and used to obtain scores for the remaining 99 pairs. We compute the Spearman correlation of the output with SimLex scores. We evaluate each of our models using tenfold cross- validation and by averaging the ten correlation figures. The changes in performance caused by previously used feature classes are described next, the performance of all major configurations are summarized in Section 4.

A. Word embeddings

Features in the first group are based on word vector similarity. For each word pair the cosine similarity of the corresponding two vectors is calculated for all embeddings used. Three sets of word vectors in our experiments were built using the neural models compared by Hill et al. (2015): the SENNA[1] (Collobert and Weston, 2008), and Huang[2] (Huang et al., 2012) embeddings contain 50-dimension vectors and were downloaded from the authors' webpages. The word2vec (Mikolov et al., 2013) vectors are of 300 dimensions and were trained on the Google News dataset[3].

We extend this set of models with GloVe vectors[4] (Pennington et al., 2014), trained on 840 billion tokens of Common Crawl data[5], and the two word embeddings mentioned in Section 1 that have recently been evaluated on the SimLex dataset: the 500-dimension SP model[6] (Schwartz et al., 2015) (see Section 1) and the 300-dimension Paragram vectors[7] (Wieting et al., 2015). The model trained on 6 features corresponding to the 6 embeddings mentioned achieves a Spearman correlation of 0.72, the performance of individual embeddings is listed in Table 1.

B. Wordnet

Another group of features are derived using WordNet (Miller, 1995). WordNet-based metrics proved to be useful in the Semeval-system of Han et al. (2013), who used these metrics for calculating a boost of word similarity scores. The top system of Banjade et al. (2015) also includes a subset of these features. We chose to use four of these metrics as binary features in our system;

[1]http://ronan.collobert.com/senna/
[2]http://www.socher.org
[3]https://code.google.com/archive/p/word2vec/
[4]http://nlp.stanford.edu/projects/glove/
[5]https://commoncrawl.org/
[6]http://www.cs.huji.ac.il/˜roys02/papers/sp_embeddings/sp_embeddings.html
[7]http://ttic.uchicago.edu/˜wieting/

System	Spearman's ρ
Huang	0.14
SENNA	0.27
GloVe	0.40
Word2Vec	0.44
SP	0.50
Paragram	0.68
6 embeddings	**0.72**

Table 1: Performance of word embeddings on SimLex

These indicate whether one word is a direct or two-link hypernym of the other, whether the two are derivationally related, and whether one word appears frequently in the glosses of the other (and its direct hypernym and its direct hyponyms). Each of these features improved our system independently; adding all of them brought the system's performance to 0.73. A model trained on the 4 WordNet-based features alone achieves a correlation of 0.33.

3 4lang

The 4lang theory of semantics was introduced and motivated in Kornai (2010) and Kornai (2012). The name refers to the initial concept dictionary, which had bindings in four languages, representative samples of the major language families spoken in Europe; Germanic (English), Slavic (Polish), Romance (Latin), and Finno-Ugric (Hungarian). Today, bindings exist in over 40 languages (Ács et al., 2013). We only present a bird's-eye view here, and refer the reader to the book-length presentation (Kornai, in preparation) for details. In brief, 4lang is an algebraic (symbolic) system that puts the emphasis on lexical definitions at the word and sub-word level, and on valency (slot-filling) on the phrase and sentence level. Paragraphs and yet higher (discourse) units are not well worked out, but these play no role in any of the approaches to analogy and similarity that we are aware of.

Historically, 4lang falls in the AI/KR tradition, following on the work of Quillian (1969), Schank (1975), and more recently Banarescu et al. (2013). Linguistically, it is closest to Wierzbicka (1972), Goddard (2002) and to modern theories of case grammar and linking theory (see Butt (2006) for a summary). Computationally, 4lang is in the finite state tradition (Koskenniemi, 1983), except it relies on an extension of finite state automata (FSA) introduced by Eilenberg (1974) to machines.

In addition to the usual state machine (where letters of the alphabet correspond to directed edges running between the states), an Eilenberg machine will also have a base set X, with each letter of the alphabet corresponding to a binary relation over X. As the machine consumes letters one by one, the corresponding relations are composed. How this mechanism can be used to account for slot- filling in a variable-free setting is described in Kornai (2010).

Central to the goals of the current paper is the structure of X. As a first approximation, X can be thought of as a hyper graph, where each hyper node is a lexeme (for a total of about 105 such hyper nodes), and hyper edges run from (hyper) node a to b if b appears in the definition of a. Since the definition of fox includes the word clever, we have a link from fox to clever, but not conversely, since the definition of clever does not refer to fox. Edges are of three types: 0, corresponding both to attribution and IS A relations; 1, corresponding to grammatical subjects; and 2, corresponding to grammatical objects. Indirect objects are handled by the decomposition methods pioneered in generative semantics, without recourse to a '3' link type (Kornai, 2012).

Each lexeme is a small Eilenberg machine, with only a few states in its FSA, so the state space X of the entire lexicon is best viewed as a large graph with about 106 states (assuming 10 states per hyper node). This base set is shared across the individual machines and functions analogously to the blackboard long familiar from AI (Nii, 1986). The primary purpose of the machine apparatus is to formalize the classical distributed model of semantic interpretation, spreading activation (Collins and Loftus, 1975; Nemeskey et al., 2013), by a series of changes in the hyper- node activation levels, described by the relations on X. Manual grammar writing in this style can lead to very high precision high recall grammars (Karlsson et al., 1995; Tapanainen and Järvinen, 1997), but for now we rely on the Stanford Parser (Chen and Manning, 2014) to produce the dependency structures that we process into simplified 4lang representations (ordinary edge-colored directed graphs rather than hyper graphs) we call definition graphs and describe briefly in Section 3.1.

We derive several similarity features from pairs of definition graphs built using the 4lang library8. Words that are not part of the manually built 4lang dictionary9 are defined by graphs built from entries in monolingual dictionaries of English using the Stanford Dependency Parser and a small hand-written mapping from dependency relations to 4lang connections (see Recski (2016) for details). The set of all words used in definitions of the Longman Dictionary of Contemporary English (Bullon, 2003), also known as the Longman Defining Vocabulary (LDV), is included in the ca. 3000 words that are defined manually in the 4lang dictionary. Recski and A´ cs (2015) used a word similarity metric based on 4lang graphs in their best STS submission, their findings served as our starting point when defining features over pairs of 4lang graphs.

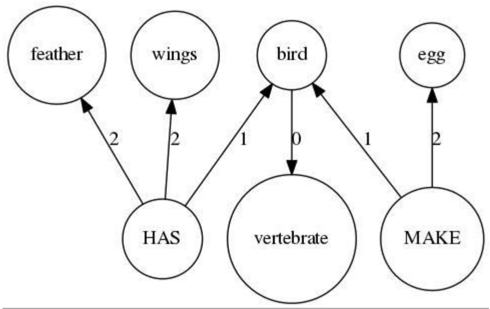

Figure 1: 4lang definition of bird.

3.1 The formalism

For the purposes of word similarity calculations we find it expedient to abstract away from some of the hypergraph/machine aspects of 4lang discussed above and represent the meaning of both words and utterances as directed graphs, similarly to the Abstract Meaning Representations (AMRs) of Banarescu et al. (2013). Nodes correspond to language-independent concepts; edges may have one of three labels (0, 1, and 2). 0-edges represent attribution (dog → friendly), the IS A relation (hypernymy) (dog → animal), and unary predication (dog 0 bark). Since concepts do not have grammatical categories, phrases like water freezes and frozen water would both be represented as water → freeze. 1- and 2-edges connect binary predicates to their arguments, e.g. cat ← catch → mouse). The meaning of each 4lang concept is represented as a 4lang graph over other concepts, e.g. the concept bird is defined by the graph in Figure 1.

3.2 Graph-based features

We experimented with various features over pairs of 4lang graphs as a source of word

[8]http://www.github.com/kornai/4lang
[9]http://hlt.bme.hu/en/resources/4lang_dict

similarity. The simple metric defined by Recski and A´ cs (2015) is based on the intuition that similar concepts will overlap in the elementary configurations they take part in: they might share a 0-neighbor, e.g. train →vehicle ←car, or they might be on the same path of 1- and 2-edges, e.g. park ←IN →town and street IN town. The metric used by Recski and A´ cs

(2015) defines the sets of predicates of each concept based on this intuition: given the example definition of bird in Figure 1, predicates of the concept bird (P (bird)) are vertebrate; (HAS, feather); (HAS, wing); (MAKE, egg). Predi- cates are also inherited via paths of 0-edges, e.g. (HAS, wing) will be a predicate of all concepts for which bird holds.

Our first feature extracted for each word pair is the Jaccard similarity of the sets of predicates of each concept, i.e.

$$\mathrm{Sim}(w1, w2) = \frac{|P(w1) \cap P(w2)|}{|P(w1) \cup P(w2)|}$$

A second similar feature takes into account all nodes accessible from each concept in its defini- tion graph. Recski and A´cs (2015) observe that this allows us to capture minor similarities between concepts, e.g. the definitions of casualty and army do not share predicates but do have a common node war (see Figure 2).

Based on boosting factors in the original met- ric we also generated three binary features. The links contain feature is true iff either concept is contained in a predicate of the other, nodes contain holds iff either concept is included in the other's definition graph, and 0 connected is true if the two nodes are con- nected by a path of 0-edges in either definition graph. All features are listed in Table 2.

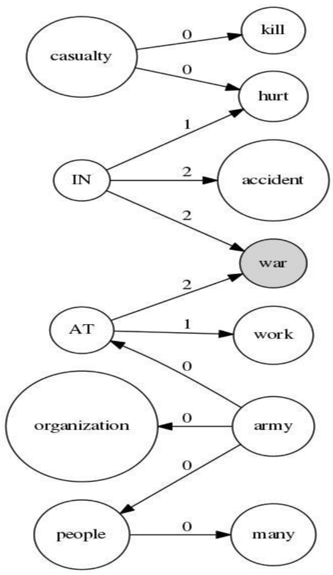

Figure 2: Overlap in the definitions of casualty (built from LDOCE) and army (defined in 4lang)

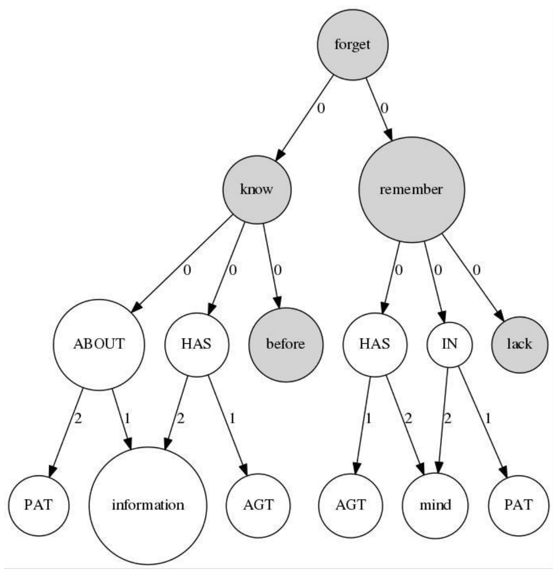

Figure 3: Expanded 4lang definition of forget. Nodes of the unexpanded graph are shown in gray.

feature	definition
links_jaccard	$J(P(w_1), P(w_2))$
nodes_jaccard	$J(N(w_1), N(w_2))$
links_contain	iff $w_1 \in P(w_2)$ or $w_2 \in P(w_1)$
nodes_contain	iff $w_1 \in N(w_2)$ or $w_2 \in N(w_1)$
0_connected	iff w_1 and w_2 are on a path of 0-edges

Table 2: 4lang word similarity features

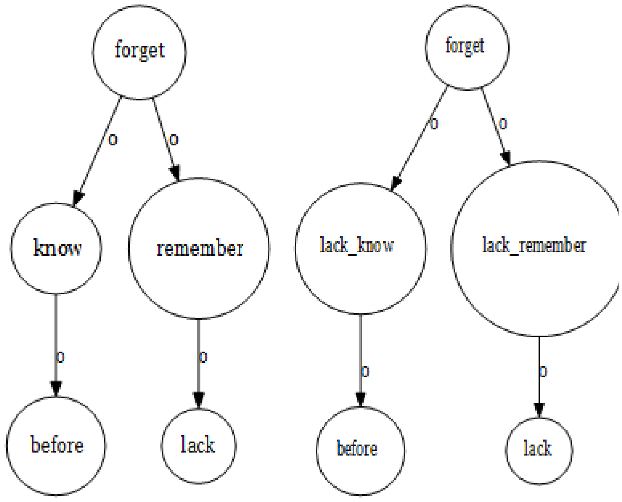

Figure 4: 4lang definition of forget and its modified version

The dict to 4lang module used to build graphs from dictionary definitions allowed us to perform expansion on each graph, which involves adjoining the definition graphs of all words to the initial graph; an example is show in Figure 3.

Using only these features in initial experi- ments resulted in many “false positives”: pairs of antonyms in SimLex were often assigned high similarity scores because this feature set is not sensitive to the 4lang nodes LACK, representing negation (dumb →−0 intelligent →−0 LACK), and BEFORE, indicating that something was only true in the past (forget →−0 know →−0 before), We attempt to model the effect of these nodes in two ways. First, we implement the is antonym feature, a binary set to true if one word is within the scope (i.e. 0-connected to) an instance of either lack or before in the other word’s graph.

Next, we transform the input graphs of remaining features so that all nodes within the scope of lack or before are prefixed by lack and are not considered identical with their non-negated counterparts when computing each of the features in Table 2. An example of such a transformation is shown in Figure 4.
Early experiments show that a system trained on 4lang-based features only can achieve a Pearson correlation in the range of 0.32 0.34 on the SimLex data, this was increased to 0.38 by the handling of LACK and BEFORE described above. This score is competitive with some word embeddings, but well below the 0.58 0.68 range achieved by the state-of-the-art vector-based systems cited in Section 1 and reproduced in Section 2.1.

After testing 4lang features’ impact on purely vector-based configurations we came to the conclusion that the only 4lang-based features that improve their performance significantly are 0-connected and is antonym. Adding these two features to the vector-based system brings correlation to 0.76.

4 Results

Performance of our main configurations is pre- sented in Table 3. The system relying on word em- beddings achieves a Spearman correlation of 0.72. WordNet and 4lang features both improve the vector-based system, combining all three feature classes yields our top correlation of 0.76, higher than any previously published results. Since the average correlation between a human rater and the average of all other raters is 0.78, this figure suggests that our system has achieved near-human performance on this benchmark.

System	Spearman's ρ
embeddings	0.72
embeddings+wordnet	0.73
embeddings+4lang	0.75
embeddings+wordnet+4lang	**0.76**

Table 3: Performance of major configurations on SimLex

For the purposes of error analysis we sorted word pairs by the difference between gold similarity values from SimLex and the output of our top-scoring model. The top of this list is clearly dominated by two error classes. The largest group consists of (near-)synonyms that have not been identified as related by our model, Table 4 shows the top 5 word pairs from this category. The second error group contains word pairs that have been falsely rewarded for being associated, but not similar by the definition used when creating the SimLex data. Table 5 shows the top 5 word pairs of this error class. This second error class is an indication of a well-known shortcoming of word similarity models: (Hill et al., 2015) observes that similarity of vectors in word embeddings tend to encode association (or relatedness) rather than the similarity of concepts.

word1	word2	output	gold	diff
bubble	suds	2.97	8.57	5.59
dense	dumb	1.71	7.27	5.56
cop	sheriff	3.50	9.05	5.55
alcohol	gin	3.43	8.65	5.22
rationalize	think	3.50	8.25	4.75

Table 4: Top 5 "false negative" errors

word1	word2	output	gold	diff
girl	maid	7.72	2.93	-4.79
happiness	luck	6.59	2.38	-4.21
crazy	sick	7.49	3.57	-3.92
arm	leg	6.74	2.88	-3.86
breakfast	supper	8.01	4.40	-3.61

Table 5: Top 5 "false positive" error

Since our main purpose was to experiment with 4lang representations and identify its shortcomings, we examined 4lang graphs of top erroneous word pairs. As expected, the value of the 0-connected feature was 1 for each "false negative" pair, i.e. word pairs such as those in Table 4 were not on the same path of 0- edges. In most cases this is due to the current lack of simple inferencing on 4lang representations. For example, suds are defined in LDOCE as the mass of bubbles formed on the top of water with soap in it, yet the resulting 4lang subgraph bubble ←1 HAS →2 mass 0 suds will not trigger any mechanism that would derive suds 0 bubble. Inference will also be responsible for deriving all uses of polysemous words, the 4lang representation of dense is therefore built from its first definition in LDOCE: made of or containing a lot of things or people that are very close together. A method of inference that will relate this definition with that of dumb is clearly out of reach. Better short-term results could be obtained by using all definitions in a dictionary to build 4lang representations, for dense this would include its third definition: not able to understand things easily.

Other shortcomings of 4lang representations are of a more technical nature, e.g. the lemmatizer used to map words of definitions to concepts failed to map alcoholic to alcohol in the definition of gin: a strong alcoholic drink made mainly from grain. Yet other errors could be addressed by rewarding the overlap between two representations, e.g. that the graphs for cop and sheriff both contain → officer.

5 Conclusions and future work

The purpose of experimenting with 4lang-based features was to gain a better understanding of how 4lang may implicitly encode semantic relations that are difficult to model with standard tools such as word embeddings or WordNet. We found that simple features describing the relation between two concepts in 4lang improve vector- based systems significantly. Since less explicit relationships may be encoded by more distant relationships in the network of 4lang concepts, in the future we plan to examine portions of this network larger than the union of two (expanded) definition graphs. Errors made by 4lang-based systems also indicate that a more sophisticated form of lexical inference on 4lang graphs may be necessary to establish the more distant connections between pairs of concepts. In the near future we plan to experiment with features defined on larger 4lang networks. We also plan to extend our system to include the task of measuring phrase similarity, which can also be pursued using supervised learning given new resources such as the Annotated-PPDB and ML-Paraphrase datasets introduced by (Wieting et al., 2015).

References

Judit A´ cs, Katalin Pajkossy, and Andra´s Kornai. 2013. Building basic vocabulary across 40 languages. In Proceedings of the Sixth Workshop on Building and Using Comparable Corpora, pages 52–58, Sofia, Bulgaria. ACL.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, In˜igo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 Task 2: Semantic Tex- tual Similarity, English, Spanish and Pilot on Inter- pretability. In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, CO, U.S.A.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In Proceedings of the 7th Linguis- tic Annotation Workshop and Interoperability with Discourse, pages 178–186, Sofia, Bulgaria. Associ- ation for Computational Linguistics.

Rajendra Banjade, Nabin Maharjan, Nobal B. Niraula, Vasile Rus, and Dipesh Gautam. 2015. Lemon and tea are not similar: Measuring word-to-word simi- larity by combining different methods. In Alexander Gelbukh, editor, Proc. CICLING15, pages 335–346. Springer.

Stephen Bullon. 2003. Longman Dictionary of Con- temporary English 4. Longman.

Miriam Butt. 2006. Theories of Case. Cambridge University Press.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural net- works. In EMNLP, pages 740–750.

A.M. Collins and E.F. Loftus. 1975. A spreading- activation theory of semantic processing. Psycho- logical Review, 82:407–428.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Pro- ceedings of the 25th International Conference on Machine Learning, ICML ’08, pages 160–167, New York, NY, USA. ACM.

Samuel Eilenberg. 1974. Automata, Languages, and Machines, volume A. Academic Press.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, , and Ey- tan Ruppin. 2002. Placing search in context: The concept revisited. ACM Transactions on Informa- tion Systems, 20(1):116–131, January.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In HLT-NAACL, pages 758–764.

Cliff Goddard. 2002. The search for the shared seman- tic core of all languages. In Cliff Goddard and Anna Wierzbicka, editors, Meaning and Universal Gram- mar – Theory and Empirical Findings, volume 1, pages 5–40. Benjamins.

Lushan Han, Abhay Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. 2013.

UMBC EBIQUITY-CORE: Semantic textual similarity systems. In Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics, pages 44–52.

Lushan Han, Justin Martineau, Doreen Cheng, and Christopher Thomas. 2015. Samsung: Align-and- Differentiate Approach to Semantic Textual Similar- ity. In Proceedings of the 9th International Work- shop on Semantic Evaluation (SemEval 2015), pages 172–177, Denver, Colorado. Association for Com- putational Linguistics.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (gen- uine) similarity estimation. Computational Linguis- tics.

Eric H. Huang, Richard Socher, Christopher D. Man- ning, and Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. In Proceedings of the 50th Annual Meet- ing of the Association for Computational Linguis- tics: Long Papers - Volume 1, ACL ’12, pages 873– 882, Stroudsburg, PA, USA. Association for Com- putational Linguistics.

Fred Karlsson, Atro Voutilainen, Juha Heikkila, and Arto Anttila. 1995. Contraint Grammar, A Language-Independent System for Parsing Unre- stricted Text. Mouton de Gruyter, Berlin, New-York.

Andra´s Kornai. 2010. The algebra of lexical seman- tics. In Christian Ebert, Gerhard Ja¨ger, and Jens Michaelis, editors, Proceedings of the 11th Mathe- matics of Language Workshop, LNAI 6149, pages 174–199. Springer.

Andra´s Kornai. 2012. Eliminating ditransitives. In Ph. de Groote and M-J Nederhof, editors, Revised and Selected Papers from the 15th and 16th Formal Grammar Conferences, LNCS 7395, pages 243–261. Springer.

Andra´s Kornai. in preparation. Semantics. http://kornai.com/Drafts/sem.pdf.

Kimmo Koskenniemi. 1983. Two-Level Morphol- ogy: A General Computational Model for Word- Form Recognition and Production. PhD thesis, Uni- versity of Helsinki.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word represen- tations in vector space. In Y. Bengio and Y. LeCun, editors, Proceedings of the ICLR 2013.

George A. Miller. 1995. Wordnet: a lexical database for English. Communications of the ACM, 38(11):39–41.

Da´vid Nemeskey, Ga´bor Recski, Ma´rton Makrai, Attila Zse´der, and Andra´s Kornai. 2013. Spreading activa- tion in language understanding. In Proc. CSIT 2013, pages 140–143, Yerevan, Armenia. Springer.

H. Penny Nii. 1986. Blackboard application systems, blackboard systems and a knowledge engineering perspective. AI Magazine, 7(3):82–110.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In Conference on Empirical Methods in Natural Language Processing (EMNLP 2014).

M. Ross Quillian. 1969. The teachable language com- prehender. Communications of the ACM, 12:459– 476.

Ga´bor Recski and Judit A´ cs. 2015. MathLingBu- dapest: Concept networks for semantic similarity. In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 543– 547, Denver, Colorado. Association for Computa- tional Linguistics.

Ga´bor Recski. 2016. Building concept graphs from monolingual dictionary entries. In Nicoletta Cal- zolari, Khalid Choukri, Thierry Declerck, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Asun- cion Moreno, Jan Odijk, and Stelios Piperidis, edi- tors, Proceedings of the Tenth International Confer- ence on Language Resources and Evaluation (LREC 2016), Portorozˇ, Slovenia. European Language Re- sources Association (ELRA).

Roger C. Schank. 1975. Conceptual Information Pro- cessing. North-Holland.

Roy Schwartz, Roi Reichart, and Ari Rappoport. 2015. Symmetric pattern based word embeddings for im- proved word similarity prediction. CoNLL 2015, page 258.

Md Arafat Sultan, Steven Bethard, and Tamara Sum- ner. 2015. DLS@CU: Sentence similarity from word alignment and semantic vector composition. In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 148– 153, Denver, Colorado. Association for Computa- tional Linguistics.

Pasi Tapanainen and Timo Ja¨rvinen. 1997. A non- projective dependency parser. In Proceedings of the 5th Conference on Applied Natural Language Pro- cessing, pages 64–71.

Anna Wierzbicka.1972. Semantic Primitives. Athena¨um, Frankfurt.

John Wieting, Mohit Bansal, Kevin Gimpel, Karen Livescu, and Dan Roth. 2015. From paraphrase database to compositional paraphrase model and back. TACL, 3:345–358.