

The S-Space Package: An Open Source Package for Word Space Models

David Jurgens

University of California, Los Angeles,
4732 Boelter Hall
Los Angeles, CA 90095
jurgens@cs.ucla.edu

Keith Stevens

University of California, Los Angeles,
4732 Boelter Hall
Los Angeles, CA 90095
kstevens@cs.ucla.edu

Abstract

We present the S-Space Package, an open source framework for developing and evaluating word space algorithms. The package implements well-known word space algorithms, such as LSA, and provides a comprehensive set of matrix utilities and data structures for extending new or existing models. The package also includes word space benchmarks for evaluation. Both algorithms and libraries are designed for high concurrency and scalability. We demonstrate the efficiency of the reference implementations and also provide their results on six benchmarks.

1 Introduction

Word similarity is an essential part of understanding natural language. Similarity enables meaningful comparisons, entailments, and is a bridge to building and extending rich ontologies for evaluating word semantics. Word space algorithms have been proposed as an automated approach for developing meaningfully comparable semantic representations based on word distributions in text.

Many of the well known algorithms, such as Latent Semantic Analysis (Landauer and Dumais, 1997) and Hyperspace Analogue to Language (Burgess and Lund, 1997), have been shown to approximate human judgements of word similarity in addition to providing computational models for other psychological and linguistic phenomena. More recent approaches have extended this approach to model phenomena such as child language acquisition (Baroni et al., 2007) or semantic priming (Jones et al., 2006). In addition, these models have provided insight in fields outside of linguistics, such as information retrieval, natural language processing and cognitive psychology. For a recent survey of word space approaches and applications, see (Cohen and Widdows, 2009).

The parallel development of word space models in different fields has often resulted in duplicated work. The pace of development presents a need for a reliable method for accurate comparisons between new and existing approaches. Furthermore, given the frequent similarity of approaches, we argue that the research community would greatly benefit from a common library and evaluation utilities for word spaces. Therefore, we introduce the **S-Space Package**, an open source framework with four main contributions:

1. reference implementations of frequently cited algorithms
2. a comprehensive, highly concurrent library of tools for building new models
3. an evaluation framework for testing models on standard benchmarks, e.g. the TOEFL Synonym Test (Landauer et al., 1998)
4. a standard interface for interacting with all word space models, which facilitates word space based applications.

The package is written in Java and defines a standardized Java interface for word space algorithms. The framework also provides reusable utilities for word spaces, such as tokenizing and filtering, sparse vectors and matrices, specialized data structures, and seamless integration with external programs for dimensionality reduction and clustering. While other word space frameworks exist, e.g. (Widdows and Ferraro, 2008), the focus of this framework is to ease the development of new algorithms and the comparison against existing models. We hope that the release of this framework will greatly facilitate other researchers in their efforts to develop and validate new word space models. The toolkit is available at <http://code.google.com/p/airhead-research/>, which includes a wiki containing detailed information on the algorithms, code documentation and mailing list archives.

2 Word Space Models

Word space models are based on the contextual distribution in which a word occurs. This approach has a long history in linguistics, starting with Firth (1957) and Harris (1968), the latter of whom defined this approach as the Distributional Hypothesis: for two words, their similarity in meaning is predicted by the similarity of the distributions of their co-occurring words. Later models have expanded the notion of co-occurrence but retain the premise that distributional similarity can be used to extract meaningful relationships between words.

Word space algorithms consist of the same core algorithmic steps: word features are extracted from a corpus and the distribution of these features is used as a basis for semantic similarity. Figure 1 illustrates the shared algorithmic structure of all the approaches, which is divided into four components: corpus processing, context selection, feature extraction and global vector space operations.

Corpus processing normalizes the input to create a more uniform set of features on which the algorithm can work. Corpus processing techniques frequently include stemming and filtering of stop words or low-frequency words. For web-gathered corpora, these steps also include removal of non linguistic tokens, such as html markup, or restricting documents to a single language.

Context selection determines which tokens in a document may be considered for features. Common approaches use a lexical distance, syntactic relation, or document co-occurrence to define the context. The various decisions for selecting the context accounts for many differences between otherwise similar approaches.

Feature extraction determines the dimensions of the vector space by selecting which tokens in the context will count as features. Features are commonly word co-occurrences, but more advanced models may perform a statistical analysis to select only those features that best distinguish word meanings. Other models approximate the full set of features to enable better scalability.

Global vector space operations are applied to the entire space once the initial word features have been computed. Common operations include altering feature weights and dimensionality reduction. These operations are designed to improve word similarity by changing the feature space itself.

Document-Based Models
LSA (Landauer and Dumais, 1997)
ESA (Gabrilovich and Markovitch, 2007)
Vector Space Model (Salton et al., 1975)

Co-occurrence Models
HAL (Burgess and Lund, 1997)
COALS (Rohde et al., 2009)

Approximation Models
Random Indexing (Sahlgren et al., 2008)
Reflective Random Indexing (Cohen et al., 2009)
TRI (Jurgens and Stevens, 2009)
BEAGLE (Jones et al., 2006)
Incremental Semantic Analysis (Baroni et al., 2007)

Word Sense Induction Models
Purandare and Pedersen (Purandare and Pedersen, 2004)
WORDSI (Jurgens and Stevens, 2010)

Table 1: Algorithms in the S-Space Package

3 The S-Space Framework

The S-Space framework is designed to be extendible, simple to use, and scalable. We achieve these goals through the use of Java interfaces, reusable word space related data structures, and support for multi-threading. Each word space algorithm is designed to run as a stand alone program and also to be used as a library class.

3.1 Reference Algorithms

The package provides reference implementations for twelve word space algorithms, which are listed in Table 1. Each algorithm is implemented in its own Java package, and all commonalities have been factored out into reusable library classes. The algorithms implement the same Java interface, which provides a consistent abstraction of the four processing stages.

We divide the algorithms into four categories based on their structural similarity: document-based, co-occurrence, approximation, and Word Sense Induction (WSI) models. Document-based models divide a corpus into discrete documents and construct the vector space from word frequencies in the documents. The documents are defined independently of the words that appear in them. Co-occurrence models build the vector space using the distribution of co-occurring words in a context, which is typically defined as a region around a word or paths rooted in a parse tree. The third category of models approximate co-occurrence data rather than model it explicitly in order to achieve better scalability for larger data sets. WSI models also use co-occurrence but also attempt to discover distinct word senses while

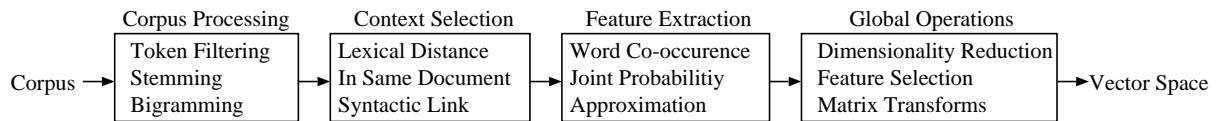


Figure 1: A high-level depiction of common algorithmic steps that convert a corpus into a word space

building the vector space. For example, these algorithms might represent “earth” with two vectors based on its meanings “planet” and “dirt.”

3.2 Data Structures and Utilities

The S-Space Package provides efficient implementations for matrices, vectors, and specialized data structures such as multi-maps and tries. These are modeled after the `java.util` library and offer concurrent implementations when multithreading is required. In addition the libraries provide support for converting between multiple matrix formats, enabling interaction with external matrix-based programs. The package also provides support for parsing different corpora formats, such as XML or email threads.

3.3 Global Operation Utilities

Many algorithms incorporate dimensionality reduction to smooth their feature data, e.g. (Landauer and Dumais, 1997; Rohde et al., 2009), or to improve efficiency, e.g. (Sahlgren et al., 2008; Jones et al., 2006). The S-Space Package supports two common techniques: the Singular Value Decomposition (SVD) and randomized projections. All matrix data structures are designed to seamlessly integrate with five SVD implementation for maximum portability: SVDLIBC¹, Matlab, Octave, JAMA² and COLT³. The package also provides a comprehensive library for randomized projections, which project high-dimensional feature data into a lower dimensional space. The library supports both integer-based projections (Kanerva et al., 2000) and Gaussian-based (Jones et al., 2006).

The package supports common matrix transformations that have been applied to word spaces: point wise mutual information (Dekang, 1998), term frequency-inverse document frequency (Salton and Buckley, 1988), and log entropy (Landauer and Dumais, 1997).

¹<http://tedlab.mit.edu/~dr/SVDLIBC/>

²<http://math.nist.gov/javanumerics/jama/>

³<http://acs.lbl.gov/~hoschek/colt/>

3.4 Measurements

The choice of similarity function for the vector space is the least standardized across approaches. Typically the function is empirically chosen based on a performance benchmark and different functions have been shown to provide application specific benefits (Weeds et al., 2004; Bullinaria and Levy, 2007). To facilitate exploration of the similarity function parameter space, the S-Space Package provides support for multiple similarity functions: cosine similarity, Euclidean distance, KL divergence, Jaccard Index, Pearson product-moment correlation, Spearman’s rank correlation, and Lin Similarity (Dekang, 1998)

3.5 Clustering

Clustering serves as a tool for building and refining word spaces. WSI algorithms, e.g. (Purandare and Pedersen, 2004), use clustering to discover the different meanings of a word in a corpus. The S-Space Package provides bindings for using the CLUTO clustering package⁴. In addition, the package provides Java implementations of Hierarchical Agglomerative Clustering, Spectral Clustering (Kannan et al., 2004), and the Gap Statistic (Tibshirani et al., 2000).

4 Benchmarks

Word space benchmarks assess the semantic content of the space through analyzing the geometric properties of the space itself. Currently used benchmarks assess the semantics by inspecting the representational similarity of word pairs. Two types of benchmarks are commonly used: word choice tests and association tests. The S-Space Package supports six tests, and has an easily extensible model for adding new tests.

4.1 Word Choice

Word choice tests provide a target word and a list of options, one of which has the desired relation to the target. Word space models solve these tests by selecting the option whose representation is most

⁴<http://glaros.dtc.umn.edu/gkhome/views/cluto>

Algorithm	Corpus	Word Choice			Word Association		
		TOEFL	ESL	RDWP	R-G	WordSim353	Deese
BEAGLE	TASA	46.03	35.56	46.99	0.431	0.342	0.235
COALS	TASA	65.33	60.42	93.02	0.572	0.478	0.388
HAL	TASA	44.00	20.83	50.00	0.173	0.180	0.318
HAL	Wiki	50.00	31.11	43.44	0.261	0.195	0.042
ISA	TASA	41.33	18.75	33.72	0.245	0.150	0.286
LSA	TASA	56.00 ^a	50.00	45.83	0.652	0.519	0.349
LSA	Wiki	60.76	54.17	59.20	0.681	0.614	0.206
P&P	TASA	34.67	20.83	31.39	0.088	-0.036	0.216
RI	TASA	42.67	27.08	34.88	0.224	0.201	0.211
RI	Wiki	68.35	31.25	40.80	0.226	0.315	0.090
RI + Perm. ^b	TASA	52.00	33.33	31.39	0.137	0.260	0.268
RRI	TASA	36.00	22.92	34.88	0.088	0.138	0.109
VSM	TASA	61.33	52.08	84.88	0.496	0.396	0.200

^a Landauer et al. (1997) report a score of 64.4 for this test, while Rohde et al. (2009) report a score of 53.4.

^b + Perm indicates that permutations were used with Random Indexing, as described in (Sahlgren et al., 2008)

Table 2: A comparison of the implemented algorithms on common evaluation benchmarks

similar. Three word choice benchmarks that measure synonymy are supported.

The first test is the widely-reported Test of English as a Foreign Language (TOEFL) synonym test from (Landauer et al., 1998), which consists of 80 multiple-choice questions with four options. The second test comes from the English as a Second Language (ESL) exam and consists of 50 question with four choices (Turney, 2001). The third consists of 200 questions from the Canadian Reader’s Digest Word Power (RDWP) (Jarmasz and Szpakowicz, 2003), which unlike the previous two tests, allows the target and options to be multi-word phrases.

4.2 Word Association

Word association tests measure the semantic relatedness of two words by comparing word space similarity with human judgements. Frequently, these tests measure synonymy; however, other types of word relations such as antonymy (“hot” and “cold”) or functional relatedness (“doctor” and “hospital”) are also possible. The S-Space Package supports three association tests.

The first test uses data gathered by Rubenstein and Goodenough (1965). To measure word similarity, word similarity scores of 51 human reviewers were gathered a set of 65 noun pairs, scored on a scale of 0 to 4. The ratings are then correlated with word space similarity scores.

Finkelstein et al. (2002) test for relatedness. 353 word pairs were rated by either 13 or 16 subjects on a 0 to 10 scale for how related the words are. This test is notably more challenging for word space models because human ratings are not tied to a specific semantic relation.

The third benchmark considers the antonym association. Deese (1964) introduced 39 antonym pairs that Greffenstette (1992) used to assess whether a word space modeled the antonymy relationship. We quantify this relationship by measuring the similarity rank of each word in an antonym pair, w_1, w_2 , i.e. w_2 is the k^{th} most-similar word to w_1 in the vector space. The antonym score is calculated as $\frac{2}{rank_{w_1}(w_2) + rank_{w_2}(w_1)}$. The score ranges from $[0, 1]$, where 1 indicates that the most similar neighbors in the space are antonyms. We report the mean score for all 39 antonyms.

5 Algorithm Analysis

The content of a word space is fundamentally dependent upon the corpus used to construct it. Moreover, algorithms which use operations such as the SVD have a limit to the corpora sizes they can process. We therefore highlight the differences in performance using two corpora. TASA is a collection of 44, 486 topical essays introduced in (Landauer and Dumais, 1997). The second corpus is built from a Nov. 11, 2009 Wikipedia snap-

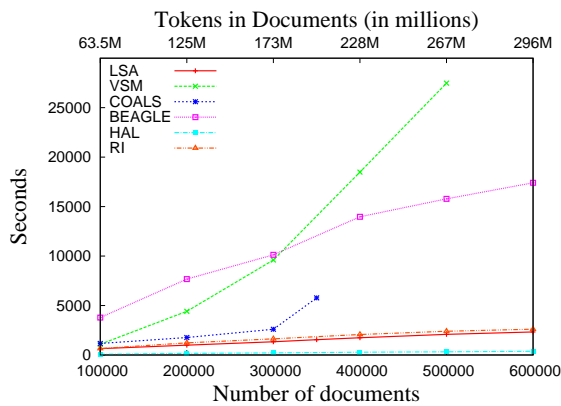


Figure 2: Processing time across different corpus sizes for a word space with the 100,000 most frequent words

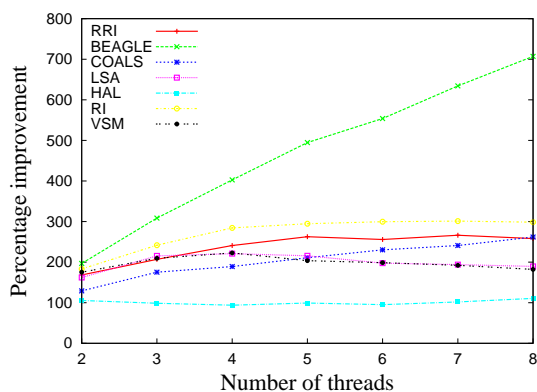


Figure 3: Run time improvement as a factor of increasing the number of threads

shot, and filtered to contain only articles with more than 1000 words. The resulting corpus consists of 387,082 documents and 917 million tokens.

Table 2 reports the scores of reference algorithms on the six benchmarks using cosine similarity. The variation in scoring illustrates that different algorithms are more effective at capturing certain semantic relations. We note that scores are likely to change for different parameter configurations of the same algorithm, e.g. token filtering or changing the number of dimensions.

As a second analysis, we report the efficiency of reference implementations by varying the corpus size and number of threads. Figure 2 reports the total amount of time each algorithm needs for processing increasingly larger segments of a web-gathered corpus when using 8 threads. In all cases, the top 100,000 words were counted as features. Figure 3 reports run time improvements due to multi-threading on the TASA corpus.

Algorithm efficiency is determined by three fac-

tors: contention on global statistics, contention on disk I/O, and memory limitations. Multi-threading benefits increase proportionally to the amount of work done per context. Memory limitations account for the largest efficiency constraint, especially as the corpus size and number of features grow. Several algorithms lack data points for larger corpora and show a sharp increase in running time in Figure 2, reflecting the point at which the models no longer fit into 8GB of memory.

6 Future Work

Recent models have used dependency parsed corpora for analysis, e.g. (Padó and Lapata, 2007). The next stage of development will be focused on providing support for dependency parsing, and other syntactic features, as well as providing reference implementations of these algorithms.

7 Conclusion

We have described a framework for developing and evaluating word space algorithms. Many well known algorithms are already provided as part of the framework as reference implementations for researches in distributional semantics. We have shown that the provided algorithms and libraries scale appropriately. Last, we motivate further research by illustrating the significant performance differences of the algorithms on six benchmarks.

References

- Marco Baroni, Alessandro Lenci, and Luca Onnis. 2007. Isa meets lara: A fully incremental word space model for cognitively plausible simulations of semantic learning. In *Proceedings of the 45th Meeting of the Association for Computational Linguistics*.
- J. A. Bullinaria and J. P. Levy. 2007. Extracting semantic representations from word co-occurrence statistics: a computational study. *Behavior Research Methods*, 39:510–526.
- Curt Burgess and Kevin Lund. 1997. Modeling parsing constraints with high-dimensional context space. *Language and Cognitive Processes*, 12:177210.
- Trevor Cohen and Dominic Widdows. 2009. Empirical distributional semantics: Methods and biomedical applications. *Journal of Biomedical Informatics*, (2):390–405.
- Trevor Cohen, Roger Schvaneveldt, and Dominic Widdows. 2009. Reflective random indexing and indirect inference: A scalable method for discovery of

- implicit connections. *Journal of Biomedical Informatics*, In Press.
- J. Deese. 1964. The associative structure of some common english adjectives. *Journal of Verbal Learning and Verbal Behavior*, 3(5):347–357.
- Lin Dekang. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the Joint Annual Meeting of the Association for Computational Linguistics and International Conference on Computational Linguistics*, pages 768–774.
- L. Finkelstein, E. Gabrilovich, Y. Matias, E. Z. S. Rivlin, G. Wolfman, and E. Ruppín. 2002. Placing search in context: The concept revisited. *ACM Transactions of Information Systems*, 20(1):116–131.
- J. R. Firth, 1957. *A synopsis of linguistic theory 1930-1955*. Oxford: Philological Society. Reprinted in F. R. Palmer (Ed.), (1968). *Selected papers of J. R. Firth 1952-1959*, London: Longman.
- Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI'07: Proceedings of the 20th international joint conference on Artificial intelligence*, pages 1606–1611.
- Gregory Grefenstette. 1992. Finding semantic similarity in raw text: The Deese antonyms. In *Working notes of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, pages 61–65. AAAI Press.
- Zellig Harris. 1968. *Mathematical Structures of Language*. Wiley, New York.
- Mario Jarmasz and Stan Szpakowicz. 2003. Roget's thesaurus and semantic similarity. In *Conference on Recent Advances in Natural Language Processing*, pages 212–219.
- Michael N. Jones, Walter Kintsch, and Douglas J. K. Mewhort. 2006. High-dimensional semantic space accounts of priming. *Journal of Memory and Language*, 55:534–552.
- David A. Jurgens and Keith Stevens. 2009. Event detection in blogs using temporal random indexing. In *Proceedings of RANLP 2009: Events in Emerging Text Types Workshop*.
- David A. Jurgens and Keith Stevens. 2010. *In submission (title suppressed)*.
- P. Kanerva, J. Kristoferson, and A. Holst. 2000. Random indexing of text samples for latent semantic analysis. In L. R. Gleitman and A. K. Josh, editors, *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, page 1036.
- Ravi Kannan, Santosh Vempala, and Adrian Vetta. 2004. On clusterings: Good, bad and spectral. *Journal of the ACM*, 51(3):497–515.
- Thomas K. Landauer and Susan T. Dumais. 1997. A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104:211–240.
- T. K. Landauer, P. W. Foltz, and D. Laham. 1998. Introduction to Latent Semantic Analysis. *Discourse Processes*, (25):259–284.
- Sebastian Padó and Mirella Lapata. 2007. Dependency-Based Construction of Semantic Space Models. *Computational Linguistics*, 33(2):161–199.
- Amruta Purandare and Ted Pedersen. 2004. Word sense discrimination by clustering contexts in vector and similarity spaces. In *HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning (CoNLL-2004)*, pages 41–48. Association for Computational Linguistics.
- Douglas L. T. Rohde, Laura M. Gonnerman, and David C. Plaut. 2009. An improved model of semantic similarity based on lexical co-occurrence. *Cognitive Science*. submitted.
- H. Rubenstein and J. B. Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8:627–633.
- M. Sahlgren, A. Holst, and P. Kanerva. 2008. Permutations as a means to encode order in word space. In *Proceedings of the 30th Annual Meeting of the Cognitive Science Society (CogSci'08)*.
- G. Salton and C. Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24:513–523.
- G. Salton, A. Wong, and C. S. Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.
- Robert Tibshirani, Guenther Walther, and Trevor Hastie. 2000. Estimating the number of clusters in a dataset via the gap statistic. *Journal Royal Statistics Society B*, 63:411–423.
- Peter D. Turney. 2001. Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the Twelfth European Conference on Machine Learning (ECML-2001)*, pages 491–502.
- Julie Weeds, David Weir, and Diana McCarty. 2004. Characterising measures of lexical distributional similarity. In *Proceedings of the 20th International Conference on Computational Linguistics COLING'04*, pages 1015–1021.
- Dominic Widdows and Kathleen Ferraro. 2008. Semantic vectors: a scalable open source package and online technology management application. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*.