EXPLORING DISTRIBUTIONAL SEMANTICS WITH THE SPACEXPLORER TOOL

ÁGOSTON TÓTH
University of Debrecen

Abstract: *Distributional studies quantify the similarity of words by collecting word co-occurrence frequency information from large text corpora. According to the distributional hypothesis, this similarity is a semantic phenomenon. This paper aims to introduce the basics of Distributional Semantics and a new tool, spaceXplorer, which facilitates distributional investigations by collecting co-occurrence information from a Wikipedia snapshot (with or without using linguistic annotation) and displays word similarity information through a convenient, interactive user interface.*
Keywords: *cognitive science, computational linguistics, distributional semantics, lexicon*

1. Introduction

Distributional Semantics is a computational, practice-oriented, data-driven approach to representing meaning. Co-occurrence statistical information supplies empirical evidence about a word's general potential for replacing another word, which gives us the opportunity of measuring word similarity. According to the distributional hypothesis, this similarity is a semantic phenomenon.

Distributional Semantics is a very active research program in Cognitive Science. It is based on a structuralist view on meaning (with roots that can be traced back to Saussure and Harris, cf. Sahlgren 2008): Distributional Semantics focuses on what is internal to language and assumes that other aspects of meaning (e.g. reference) will also be reflected by language-internal phenomena or remain irrelevant for description. Approximating the meaning of words is carried out by assessing distributional properties as manifested in corpora.

A geometric procedure is commonly employed in Distributional Semantics to represent and compare meanings. Co-occurrence events between words are usually collected as numerical features in *feature vectors* that stand for words in a *vector space*. Meaning differences and similarities can then be conveniently represented and calculated in this vector space by working with the feature vectors. More details about this process will be provided in section 2.

As shown above, Distributional Semantics is bound with strong ties to Linguistics and Geometry. Computational linguists have also found the distributional methodology an efficient yet powerful way of acquiring semantic information about words. As far as language technology is concerned, some of the first vector-space applications included the task of finding relevant documents in Information Retrieval (Salton 1971). Question answering (e.g. Tellex et al. 2003) and document clustering (e.g. Manning et al. 2008) may be implemented in a similar way. Comparable systems have been developed for word sense disambiguation (Schütze 1998), thesaurus generation through automatized discovery and clustering of word senses (Crouch 1988, Pantel and Lin 2002) and

named-entity recognition (Vyas and Pantel 2009). Pennacchiotti et al. (2008) use Distributional Semantics in a cognitive semantic context: they propose a method for extending FrameNet's scope by covering more (potentially: frame-evoking) lexical items through distributional lexical unit induction.

Psycholinguistics also has a major role it Distributional Semantics as corpus-derived and psycholinguistic data correlate (gained from human similarity judgements, cf. e.g. Miller and Charles 1991, and from semantic priming experiments, e.g. Pado and Lapata 2007).

Distributional Semantics is a powerful model that has been used in many scientific disciplines, but it has an empirical side that can only be researched with proper tools that can process large corpora and find co-occurrence events between words.

2. How to build a Vector Space Model (VSM)?

Systems designed to collect distributional information about words rely on a geometrical interpretation of the empirical data (Widdows 2004). Each target word is represented in a multi-dimensional space by a feature vector. Each position of the feature vector signals or counts the number of co-occurrences of the given target word with one of the context words we use for describing target items. For example, if the word *drink* is a target word, the word *tea* is among the context words and *tea* occurs 23 times in the close vicinity (in the "context window") of *drink*, then the vector element corresponding to the word *tea* (in the context vector describing the word *drink*) will be set to 23:

$$v_{drink} = < freq_1, freq_2, \ldots, 23, \ldots, freq_t >$$

where *v* is a feature vector that represents a target word in a *t*-dimensional space; *t* stands for the total number of context words.

Large corpora (20-50-100 million words or even more) are necessary for this type of investigation. "Raw", unprocessed corpora may be suitable for the task. In the presence of linguistic annotation, we can take additional details into consideration.

We can compare the distributions of the target words by carrying out calculations with their feature vectors. Here, I will limit the discussion to two basic methods for comparing vectors: we can measure vector distances (Figure 1) or the cosine of the angle between vectors (Figure 2). The latter promises the advantage of being able to avoid problems arising from vector length differences, which is useful, since length depends on the frequency of context words.

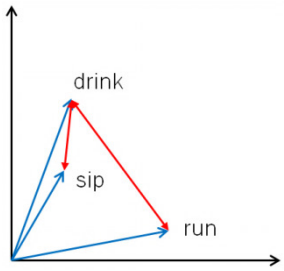

Figure 1. Vector similarity: distance

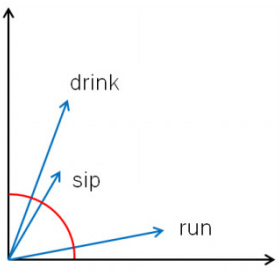

Figure 2. Vector similarity: cosine

3. The spaceXplorer program

Unlike 1st-order word co-occurrence phenomena (seen when words occur together, e.g. in idioms, compounds with open spelling, phrasal verbs, light verbs, collocations, etc.) which are possible to spot by collecting examples of words and listing them (a task readily supported by most concordancing programs), 2nd-order word co-occurrence phenomena, which are observed and quantified in distributional studies, are not directly visible to the naked eye. We need software tools that process text, find occurrences of target words and collect and statistically evaluate contextual information about each occurrence in context so that the degree to which *words occur with the same words* can be calculated.

3.1. Measuring word similarity using spaceXplorer

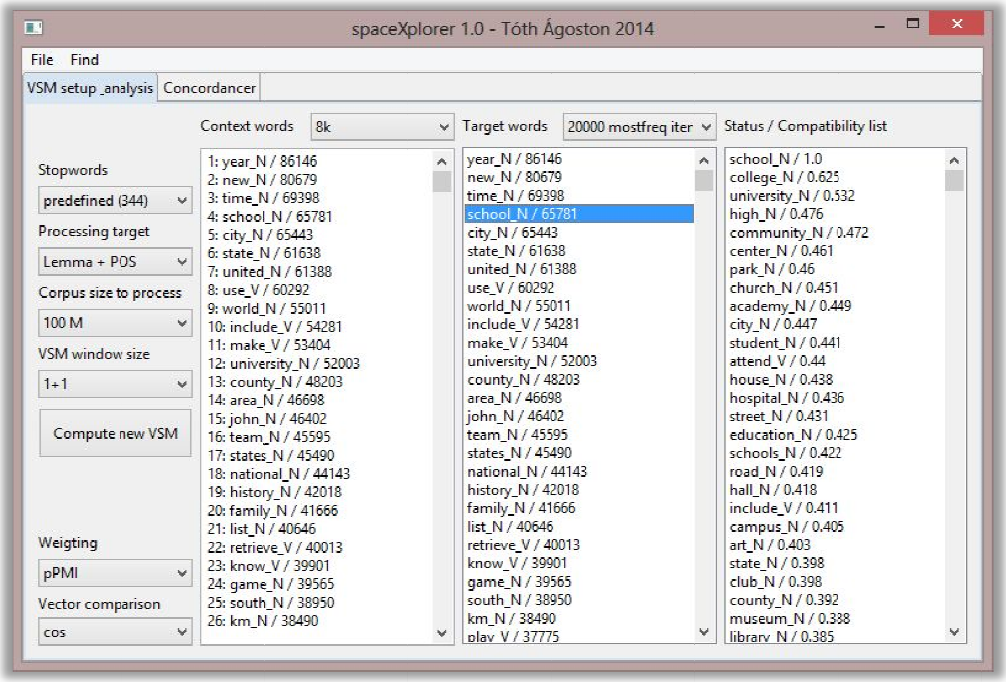

Figure 3

SpaceXplorer (Figure 3) is a text processing tool for distributional studies. Before we start processing the corpus, the following parameters must be set from a list of options available on the left side of the interface shown in Figure 3:

- stopwords*: whether we want to exclude the most common (function) words from processing,*
- processing target*: lemma or word form with or without POS labels,*
- corpus size to process*: up to 100 million words (using 10,000-word samples from the Wikipedia snapshot),*
- VSM window size: *the size of the moving window around the target word where the context words are detected.*

Near the top of the interface, we get options for selecting the amount of *target words* and *context words* for processing. The corresponding lists of target and context words become visible immediately, together with corpus frequency data, as shown in the *Context words* and *Target words* fields of Figure 3.

The present form of the program is prepared to process the TC Wikipedia ("Tagged and Cleaned Wikipedia") corpus available at <http://nlp.cs.nyu.edu/wikipedia-data/>.

Having set all these parameters, the user can start processing the corpus and collecting statistical information about the target words by pressing the "Compute new VSM" button. Processing time depends mainly on the amount of context words, target words and the corpus size. Running the program at its most time consuming settings requires about 24 hours on a single PC to complete. It can finish in much less time (sometimes in minutes) with more modest settings, which is useful for parameter tuning or when the program is run for demonstration purposes only. Crucially, the user can save the results and load them later; in this way, multiple investigations can be carried out without having to process the corpus again with the same parameters.

When corpus processing is over, we can select any target word from the *Target words* list: as a result, the *Compatibility list* field displays all (target) words in decreasing order of distributional similarity to the selected target. The degree of similarity is also displayed in the list. In Figure 3, for instance, the noun *school* is selected for analysis. The noun *school* appears at the top of the similarity list (compatibility score: 1.0), followed by the noun *college* (compatibility: 0.625) and the noun *university* (compatibility: 0.532). Compatibility scores are all relative to the distribution of the selected target word (*school_N*) and they show the level of freedom (0-1) with which a word can replace another word in the corpus using the selected parameter set. For calculating the compatibility scores and ordering the list, a similarity measure and a weighting scheme must also be selected on the interface (in this case: *pPMI weighting* and *cosine similarity*; you can find these settings in the bottom left corner). These options can be freely changed at any time without having to recalculate the word-context matrix by parsing the corpus again.

3.2. Creating concordances

A concordancer locates occurrences of a search expression and lists them in context. Concordance listings can be used in general linguistic research, lexicography and language learning. Concordances were used before electronic computers were invented in the middle of the 20th century; the first concordances were compiled for the Bible. In many cases, producing a concordance took years or even decades, and the result was published in several volumes. The invention of electronic computers sped up the process of compiling a concordance considerably. The spread of home computers also made the process interactive: you do not need to print the concordance, but you can generate it on the spot. Moreover, you can sort the concordance or change the options so that the pattern you are seeking becomes readily observable.

With spaceXplorer, you can create a concordance listing for the words selected from the *Compatibility list* field, which is filled with data during a VSM experiment. The concordance function can only be accessed after a VSM exploration is complete.

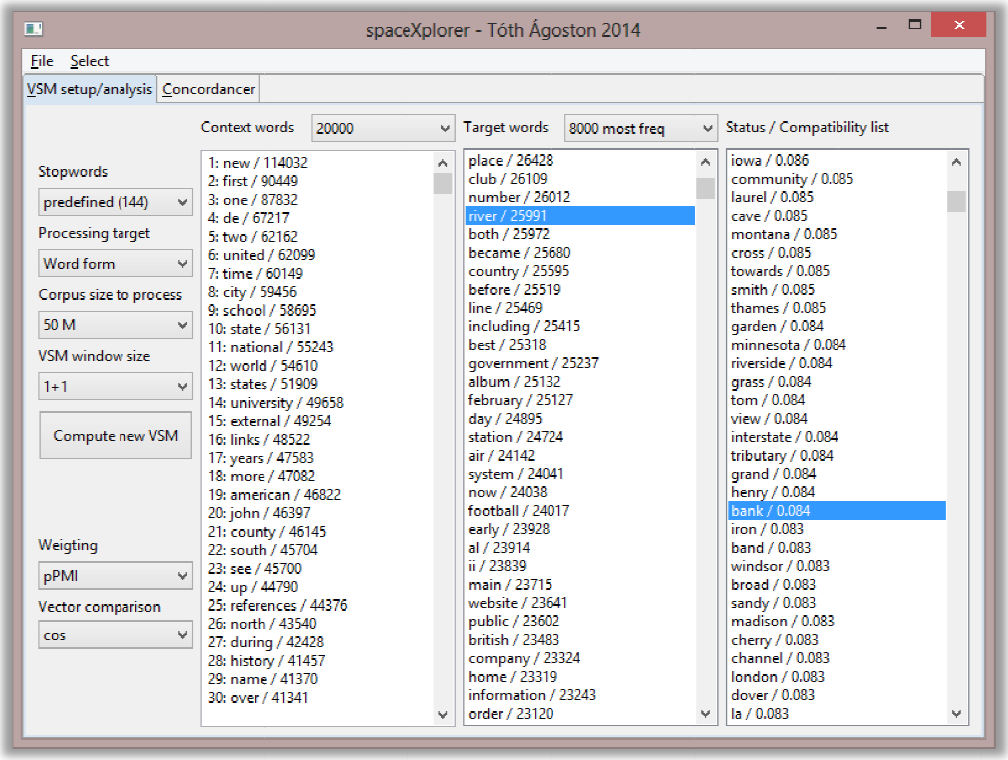

Figure 4

Figure 4 shows a compatibility list generated for the target word *river* by processing 50 million words (using no lemmatization and no POS labels at this time). After the analysis, the target words (the 8000 most frequent words of the corpus) are listed in descending order of compatibility with *river* in the *Compatibility list* field. Note that you do not see the beginning of the list in Figure 4, as the word *bank* has already been selected for further processing in this case. Also note that you do not have to browse the list manually: the program can help you locate target words and compatibles by using the corresponding functions on the *Select* menu.

When you activate the concordancer function by clicking on the *Concordancer* tab on the screen, you will get a *keyword in context* concordance shown in Figure 5. The major output elements are tabulated into columns:

- *left context,*
- *keyword (*bank*),*
- *right context,*
- *degree of relevance (0.0 – 1.0),*
- *context words available in the given sentence for calculating relevance.*

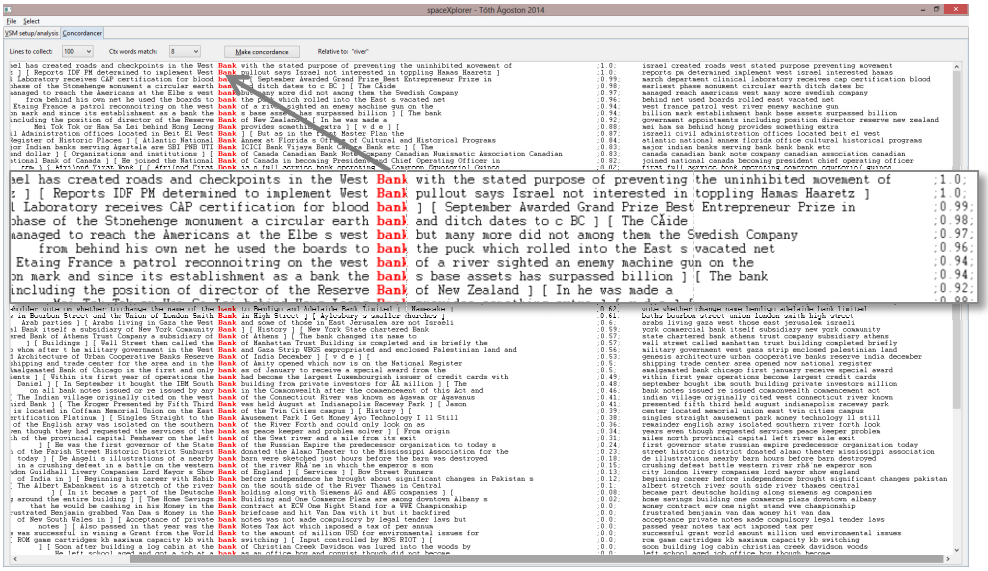

Figure 5

The concordance function comes with its own set of parameters. Most importantly, the user must specify the number of context words (used in the VSM-based part of the experiment) for calculating *relevance*. Theoretically, sentences with higher relevance score tolerate a change from the concordance keyword (in this case: *bank*) to the target word selected in the VSM experiment (here: *river*).

In this example, the following four sentences are returned as the most relevant ones on the basis of the Wikipedia data (relevance scores are specified in parentheses):

(1) *Israel has created roads and checkpoints in the West Bank with the stated purpose of preventing the uninhibited movement of… (1.0)*
(2) *Reports IDF PM determined to implement West Bank pullout says Israel not interested in toppling Hamas… (1.0)*
(3) *… Laboratory receives CAP certification for blood bank. (0.99)*
(4) *The earliest phase of the Stonehenge monument a circular earth bank and ditch dates to # BC… (0.98)*

Bank is a well-known example of homonymy (largely because of its "financial institution" and "river bank" senses). In this particular example, the program was meant to return examples with the "river bank" meaning, since the program was asked to locate sentences in which *river* and *bank* were interchangeable.

Sentences (1) and (2) seem to be correct at first sight. Unfortunately, sentences (1) - (3) are all affected by tokenization issues, the problem of chunking text into appropriate units. *West Bank* in sentence (1) and (2) is a proper name which should have been handled as a single word. The spaceXplorer program received no information about this tokenization option from the corpus (this is a very common problem with multi-word entities). Sentence (3) is similar: *blood bank* is a compound; therefore, it should have been handled as a single lexical item. Sentence (4) is unaffected by these tokenization problems and delivers a sense very

close to the expected "river bank" sense. Please note that the majority of the sentences listed in Figure 5 contain the "financial institution" sense as it is much more frequent (in the Wikipedia subcorpus being used) than the "river bank" sense.

Distributional Semantics collects data about *word types;* characterizing individual tokens (occurrences of a word) is a rarity in the literature. Some notable exceptions are Reisinger and Mooney's (2010) prototype-based approach, Erk and Padó's (2010) exemplars and Scheible, Schulte im Walde and Springorum's (2013) *Codis-Contexts* disambiguation method. They tackled the question of lexical ambiguity (in very different ways) within the framework of Distributional Semantics. The spaceXplorer program offers a traditional, *word type*-based approach to measuring distributional compatibility, without provisions for disambiguation. Therefore, distributional data remains fully affected by lexical ambiguity. The concordance function of spaceXplorer lets the user pick out authentic examples of use, while quantifying relevance of each corpus sentence as to the compatibility relation between the selected target word and the selected compatible(s); the user can browse the list and work with the examples that he or she prefers.

As shown above, the proper analysis and interpretation of corpus data continue to require human linguistic knowledge and intuition. In this respect, *distributional concordancing* is not different from traditional concordancing, where frequency, t-score, MI-score etc. information give assistance for the user to find collocations (by creating lists of potential collocations), but it is up to the human user to find the appropriate patterns.

4. Concluding remarks

The spaceXplorer program introduced in this paper is a clean, easily available, free distributional semantic tool, suitable for many situations. The program uses a Wikipedia snapshot and supports linguistic annotation (lemma and POS information). It lets the user carry out corpus-based word similarity experiments and it offers a concordancing facility.

The present form of the program is not suitable for massive qualitative evaluations, compositional distributional studies or for creating large word-context matrices. It does, however, support small and medium-size investigations, qualitative and quantitative studies, run on Windows computers, offer a graphical user interface (GUI) and let the user set the processing parameters easily and see the effect on the screen. It is also designed to be an ideal tool for teaching distributional semantics to students of linguistics and psycholinguistics. The program is available for free from the author of this paper (toth.agoston@arts.unideb.hu).

Acknowledgement: This research was supported by the European Union and the State of Hungary, co-financed by the European Social Fund in the framework of TÁMOP-4.2.4.A/ 2-11/1-2012-0001'National Excellence Program'.

References

Crouch, C. J. 1988. 'A Cluster-based Approach to Thesaurus Construction' in Y. Chiaramella (ed.). *SIGIR'88, Proceedings of the 11th Annual International ACM SIGIR Conference*. New York: Association for Computational Linguistics, pp. 309-320.

Erk, K. and S. Padó, 2010. 'Exemplar-based Models for Word Meaning in Context' in J. Hajic, S. Carberry, S. Clark and J. Nivre (eds.). *Proceedings of ACL 2010 Conference Short Papers.* Uppsala: Association for Computational Linguistics, pp. 92 – 97.

Manning, C. D., P. Raghavan, H. Schütze. 2008. *Introduction to Information Retrieval*. Cambridge, Cambridge University Press.

Miller, G. A. and W. G. Charles, 1991. 'Contextual Correlates of Semantic Similarity' in *Language and Cognitive Processes*, vol. 6(1), pp. 1-28.

Pado, S. and M. Lapata. 2007. 'Dependency-based Construction of Semantic Space Models" in *Computational Linguistics*, vol. 33(2), pp. 161-199.

Pantel, P. and D. Lin, 2002. 'Discovering Word Senses from Text' in O. R. Zaïane, R. Goebel, D. Hand, D. Keim, R. Ng (eds.). *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM, pp. 613-619.

Pennacchiotti, M., D. D. Cao, R. Basili, D. Croce, M. Roth 2008. 'Automatic Induction of FrameNet Lexical Units' in M. Lapata and H. Tou Ng. (eds.). *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP-08)*, pp. 457-465.

Reisinger, J. and R. J. Mooney, 2010. 'Multi-Prototype Vector-Space Models of Word VyMeaning' in D. Cer, Stroudsburg, Pennsylvania, Association for Computational Linguistics, C. D. Manning, D. Jurafsky (eds.). *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Proceedings NAACL*. Stroudsburg, Pennsylvania: Association for Computational Linguistics, pp. 109-117.

Sahlgren, M. 2008. 'The Distributional Hypothesis' in *Rivista di Linguistica (Italian Journal of Linguistics)*, vol. 20(1), pp. 33-53.

Salton, G. 1971. *The SMART Retrieval System: Experiments in Automatic Document Processing*. Upper Saddle River, New Jersey: Prentice-Hall.

Scheible, S., S. Schulte im Walde, S. Springorum 2013. 'Uncovering Distributional Differences between Synonyms and Antonyms in a Word Space Model' in *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pp. 489-497. [Online]. Available: http://www.aclweb.org/anthology/I/I13/. [Accessed 2014, May 15].

Schütze, H. 1998. 'Automatic Word Sense Discrimination' in *Computational Linguistics*, vol. 24(1), pp. 97-124.

Tellex, S., B., Katz, J., Lin, A. Fern, G. Marton 2003. 'Quantitative Evaluation of Passage Retrieval Algorithms for Question Answering' in J. Callan, G. Cormack, C. Clarke, D. Hawking, A. Smeaton (eds.). *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR).* Toronto: ACM Press, pp. 41-47.

Vyas, V. and P. Pantel 2009. 'Semi-automatic Entity Set Refinement' in U. Germann, C. Shah, S. Stoyanchev, C. P. Rose, A. Sarkar (eds.). *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, NAACL-09*, pp. 290-298.

Widdows, D. 2004. *Geometry and Meaning*. Stanford, California: Center for the Study of Language and Information.