Theory on Vector Space (VS)

Nikita Aneja
Ganga Technical Campus, Ganga Group of Institutions, Desalpur, Haryana, India

International Journal of Research in Mathematics & Computation
Volume 3, Issue 2, July-December, 2015, pp. 10-14
ISSN Online: 2348-1528, Print: 2348-151X, DOA : 29102015
© IASTER 2015, www.iaster.com

ABSTRACT

Computers understand very little of the meaning of human language. This profoundly limits our ability to give instructions to computers, the ability of computers to explain their actions to us, and the ability of computers to analyse and process text. Vector space models (VSMs) of semantics are beginning to address these limits. This paper surveys the use of VSMs for semantic processing of text. We organize the literature on VSMs according to the structure of the matrix in a VSMs. These are currently three broad classes of VSMs, based on term-document, word-context. And pair-pattern matrices, yielding three classes of application.

I. INTRODUCTION

One of the biggest obstacles to making full use of the power of computers is that they currently understand very little of the meaning of human language. Recent progress in search engine technology is only scratching the surface of human language, and yet the impact on society and the economy is already immense. This hints at the transformative impact that deeper semantic technologies will have. Vector space models (VSMs), surveyed in this paper, are likely to be a part of these new semantic technologies. In this paper, we use the term semantics in a general sense, as the meaning of a word, a phrase, a sentence, or any text in human language, and the study of such meaning. We are not concerned with narrower senses of semantics, such as the semantic web or approaches to semantics based on formal logic. We present a survey of VSMs and their relation with the distributional hypothesis as an approach to representing some aspects of natural language semantics.

II. MOTIVATION FOR VECTOR SPACE MODELS OF SEMANTICS

The VSM was developed for the SMART information retrieval system (Salton, 1971) by Gerard Salton and his colleagues (Salton, Wong, & Yang, 1975). SMART pioneered many of the concepts that are used in modern search engines (Manning, Raghavan, & Sch¨utze, 2008). The idea of the VSM is to represent each document in a collection as a point in a space (a vector in a vector space). Points that are close together in this space are semantically similar and points that are far apart are semantically distant. The user's query is represented as a point in the same space as the documents (the query is a *pseudo document*). The documents are sorted in order of increasing distance (decreasing semantic similarity) from the query and then presented to the user.

The success of the VSM for information retrieval has inspired researchers to extend the VSM to other semantic tasks in natural language processing, with impressive results. For instance, Rapp (2003) used a vector-based representation of word meaning to achieve a score of 92.5% on multiple-choice synonym questions from the Test of English as a Foreign Language (TOEFL), whereas the average human score was 64.5%. Turney (2006) used a vector-based representation of semantic relations to

 ISSN (O) 2348-1528
(P) 2348-151X

attain a score of 56% on multiple-choice analogy questions from the SAT college entrance test, compared to an average human score of 57%. In this survey, we have organized past work with VSMs according to the type of matrix involved: term–document, word–context, and pair–pattern. We believe that the choice of a particular matrix type is more fundamental than other choices, such as the particular linguistic processing or mathematical processing.

III. VECTOR SPACE MODELS OF SEMANTICS

The theme that unites the various forms of VSMs that we discuss in this paper can be stated as the *statistical semantics hypothesis*: statistical patterns of human word usage can be used to figure out what people mean.9 This general hypothesis underlies several more specific hypotheses, such as the *bag of words hypothesis*, the *distributional hypothesis*, the *extended distributional hypothesis*, and the *latent relation hypothesis*, discussed below.

A. The Term–Document Matrix

In this paper, we use the following notational conventions: Matrices are denoted by bold capital letters, A. Vectors are denoted by bold lowercase letters, b. Scalars are represented by lowercase italic letters, *c*. If we have a large collection of documents, and hence a large number of document vectors, it is convenient to organize the vectors into a matrix. The row vectors of the matrix correspond to terms (usually terms are words, but we will discuss some other possibilities) and the column vectors correspond to documents (web pages, for example). This kind of matrix is called a *term–document* matrix.

B. The Word–Context Matrix

Salton et al. (1975) focused on measuring document similarity, treating a query to a search engine as a pseudo-document. The relevance of a document to a query is given by the similarity of their vectors. Deerwester et al. (1990) observed that we can shift the focus to measuring word similarity, instead of document similarity, by looking at row vectors in the term–document matrix, instead of column vectors. Deerwester et al. (1990) were inspired by the term–document matrix of Salton et al. (1975), but a document is not necessarily the optimal length of text for measuring word similarity. In general, we may have a *word–context* matrix, in which the context is given by words, phrases, sentences, paragraphs, chapters, documents, or more exotic possibilities, such as sequences of characters or patterns.

C. The Pair–Pattern Matrix

In a pair–pattern matrix, row vectors correspond to pairs of words, such as mason : stone and carpenter : wood, and column vectors correspond to the patterns in which the pairs co occur, such as “X cuts Y ” and “X works with Y ”. Lin and Pantel (2001) introduced the pair–pattern matrix for the purpose of measuring the semantic similarity of patterns; that is, the similarity of column vectors. Given a pattern such as “X solves Y ”, their algorithm was able to find similar patterns, such as “Y is solved by X”, “Y is resolved in X”, and “X resolves Y ”.

IV. SIMILARITIES

It is tempting to suppose that relational similarity can be reduced to attributional similarity. For example, *mason* and *carpenter* are similar words and *stone* and *wood* are similar words; therefore, perhaps it follows that *mason* : *stone* and *carpenter* : *wood* have similar relations. Perhaps simr (*a*: *b*, *c* :*d*) can be reduced to sima (*a*, *c*)+sima(*b*, *d*). However, *mason*, *carpenter*, *potter*, and *glassblower* are similar words (they are all artisans), as are *wood*, *clay*, *stone*, and *glass* (they are all materials used

 ISSN (O) 2348-1528
(P) 2348-151X

by artisans), but we cannot infer from this that *mason* : *glass* and *carpenter* : *clay* have similar relations. Turney (2006, 2008a) presented experimental evidence that relational similarity does not reduce to attributional similarity.

In computational linguistics, the term *semantic similarity* is applied to words that share a hypernym (*car* and *bicycle* are semantically similar, because they share the hypernym *vehicle*) (Resnik, 1995). Semantic similarity is a specific type of attributional similarity. We prefer the term *taxonomical similarity* to the term *semantic similarity*, because the term *semantic similarity* is misleading. Intuitively, both attributional and relational similarity involve meaning, so both deserve to be called semantic similarity.

V. LINGUISTIC PROCESSING FOR VECTOR SPACE MODELS

We will assume that our raw data is a large corpus of natural language text. Before we generate a term–document, word–context, or pair–pattern matrix, it can be useful to apply some linguistic processing to the raw text. The types of processing that are used can be grouped into three classes. First, we need to tokenize the raw text; that is, we need to decide what constitutes a term and how to extract terms from raw text. Second, we may want to normalize the raw text, to convert superficially different strings of characters to the same form (e.g., car, Car, cars, and Cars could all be normalized to car). Third, we may want to annotate the raw text, to mark identical strings of characters as being different (e.g., fly as a verb could be annotated as fly/VB and fly as a noun could be annotated as fly/NN).

VI. MATHEMATICAL PROCESSING FOR VECTOR SPACE MODELS

The first step is to generate a matrix of frequencies. Second, we may want to adjust the weights of the elements in the matrix, because common words will have high frequencies, yet they are less informative than rare words. Third, we may want to smooth the matrix, to reduce the amount of random noise and to fill in some of the zero elements in a sparse matrix. Fourth, there are many different ways to measure the similarity of two vectors. Lowe (2001) gives a good summary of mathematical processing for word–context VSMs. He decomposes VSM construction into a similar four-step process: calculate the frequencies, transform the raw frequency counts, smooth the space (dimensionality reduction), then calculate the similarities.

VII. BUILDING THE FREQUENCY MATRIX

An element in a frequency matrix corresponds to an *event*: a certain item (term, word, word pair) occurred in a certain situation (document, context, pattern) a certain number of times (frequency). At an abstract level, building a frequency matrix is a simple matter of counting events. In practice, it can be complicated when the corpus is large. A typical approach to building a frequency matrix involves two steps. First, scan sequentially through the corpus, recording events and their frequencies in a hash table, a database, or a search engine index. Second, use the resulting data structure to generate the frequency matrix, with a sparse matrix representation (Gilbert, Moler, & Schreiber, 1992).

VIII. THE FUTURE OF VECTOR SPACE MODELS OF SEMANTICS

Several authors have criticized VSMs (French & Labiouse, 2002; Pad´o & Lapata, 2003; Morris & Hirst, 2004; Budanitsky & Hirst, 2006). Most of the criticism stems from the fact that term–document and word–context matrices typically ignore word order. In LSA, for instance, a phrase is commonly represented by the sum of the vectors for the individual words in the phrase; hence the phrases *house boat* and *boat house* will be represented by the same vector, although they have different meanings. In

 ISSN (O) 2348-1528
(P) 2348-151X

English, word order expresses relational information. Both *house boat* and *boat house* have a Tool-Purpose relation, but *house boat* means Tool-Purpose(*boat, house*) (a boat that serves as a house), whereas *boat house* means Tool-Purpose(*house, boat*) (a house for sheltering and storing boats). Landauer (2002) estimates that 80% of the meaning of English text comes from word choice and the remaining 20% comes from word order. However, VSMs are not inherently limited to 80% of the meaning of text. Mitchell and Lapata (2008) propose composition models sensitive to word order. For example, to make a simple additive model become syntax-aware, they allow for different weightings of the contributions of the vector components. Constituents that are more important to the composition therefore can participate more actively. Clark and Pulman (2007) assigned distributional meaning to sentences using the Hilbert space tensor product. Widdows and Ferraro (2008), inspired by quantum mechanics, explores several operators for modeling composition of meaning. Pair–pattern matrices are sensitive to the order of the words in a pair (Turney, 2006). Thus there are several ways to handle word order in VSMs. This raises the question, what are the limits of VSMs for semantics? Can all semantics be represented with VSMs? There is much that we do not yet know how to represent with VSMs. For example, Widdows (2004) and van Rijsbergen (2004) show how disjunction, conjunction, and negation can be represented with vectors, but we do not yet know how to represent arbitrary statements in first-order predicate calculus. However, it seems possible that future work may discover answers to these limitations.

CONCLUSIONS

When we want information or help from a person, we use words to make a request or describe a problem, and the person replies with words. Unfortunately, computers do not understand human language, so we are forced to use artificial languages and unnatural user interfaces. In science fiction, we dream of computers that understand human language, that can listen to us and talk with us. To achieve the full potential of computers, we must enable them to understand the semantics of natural language. VSMs are likely to be part of the solution to the problem of computing semantics.

Many researchers who have struggled with the problem of semantics have come to the conclusion that the meaning of words is closely connected to the statistics of word usage . When we try to make this intuition precise, we soon find we are working with vectors of values derived from event frequencies; that is, we are dealing with VSMs. In this survey, we have organized past work with VSMs according to the structure of the matrix (term–document, word–context, or pair–pattern). We believe that the structure of the matrix is the most important factor in determining the types of applications that are possible. The linguistic processing and mathematical processing play smaller (but important) roles. Our goal in this survey has been to show the breadth and power of VSMs, to introduce VSMs to those who less familiar with them, and to provide a new perspective on VSMs to those who are already familiar with them. We hope that our emphasis on the structure of the matrix will inspire new research. There is no reason to believe that the three matrix types we present here exhaust the possibilities. We expect new matrix types and new tensors will open up more applications for VSMs. It seems possible to us that all of the semantics of human language might one day be captured in some kind of VSM.

REFERENCES

[1]. Acar, E., & Yener, B. (2009). Unsupervised Multiway Data Analysis: A Literature Survey. *IEEE Transactions on Knowledge and Data Engineering*, *21* (1), 6–20.

[2]. Agirre, E., & Edmonds, P. G. (2006). *Word Sense Disambiguation: Algorithms and Applications*, Springer.

ISSN (O) 2348-1528
(P) 2348-151X

[3]. Ando, R. K. (2000). Latent semantic space: Iterative Scaling Improves Precision of Interdocument Similarity Measurement. In *Proceedings of the 23rd Annual ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-2000)*, pp.216–223.

[4]. Ando, R. K., & Zhang, T. (2005). A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data. *Journal of Machine Learning Research*, *6*, 1817–1853.

[5]. Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison Wesley.

[6]. Barr, C., Jones, R., & Regelson, M. (2008). The Linguistic Structure of English Web Search Queries. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[7]. Bayardo, R. J., Ma, Y., & Srikant, R. (2007). Scaling up all Pairs Similarity Search. In *Proceedings of the 16th international conference on World Wide Web (WWW '07)*, pp. 131–140, New York, NY. ACM.

[8]. Bi, cici, E., & Yuret, D. (2006). Clustering Word Pairs to Answer Analogy Questions. In *Proceedings of the Fifteenth Turkish Symposium on Artificial Intelligence and NeuralNetworks (TAINN 2006)*, Akyaka, Mugla, Turkey.

[9]. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

[10]. Brand, M. (2006). Fast Low-Rank Modifications of the thin Singular Value Decomposition. *Linear Algebra and Its Applications*, *415* (1), 20–30.

[11]. Breese, J., Heckerman, D., & Kadie, C. (1998). Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proceedings of the 14th Conference on Uncertainty inArtificial Intelligence*, pp. 43–52. Morgan Kaufmann.

[12]. Brin, S., & Page, L. (1998). The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *Proceedings of the Seventh World Wide Web Conference (WWW7)*, pp. 107–117.

[13]. Broder, A. (1997). On the Resemblance and Containment of Documents. *In Compression and Complexity of Sequences (SEQUENCES'97)*, pp. 21–29. IEEE Computer Society.