Natural Language Engineering **23** (1): 155–162.

doi:10.1017/S1351324916000334

Emerging Trends

Word2Vec

KENNETH WARD CHURCH
IBM, Yorktown Heights, NY, USA
e-mail: kwchurch@us.ibm.com

(*Accepted 29 August 2016*)

Abstract

My last column ended with some comments about Kuhn and word2vec. Word2vec has racked up plenty of citations because it satisifies both of Kuhn's conditions for emerging trends: (1) a few initial (promising, if not convincing) successes that motivate early adopters (students) to do more, as well as (2) leaving plenty of room for early adopters to contribute and benefit by doing so. The fact that Google has so much to say on 'How does word2vec work' makes it clear that the definitive answer to that question has yet to be written. It also helps citation counts to distribute code and data to make it that much easier for the next generation to take advantage of the opportunities (and cite your work in the process).

1 Why are some papers cited more than others?

Plenty has been written about word2vec and plenty more will be written in the future.[1] Given that reality, as well as a severe page limit, there is little hope that I could say much here that hasn't been said already. My point here is not to praise word2vec or bury it, but to discuss the discussion. Are there lessons to be learned about how to rack up citation counts?

What kinds of papers are massively cited? The definitive last word on a subject? Probably not. The first paper on a subject is more likely to be cited than the last word. That said, while originality is appreciated, the most cited paper is often not the first, or the last, or even the best. Free online availability substantially increases a paper's impact (Lawrence 2001; Eysenbach 2006). Simplicity and accessibility are preferred over timing and accuracy.

[1] See http://www.slideshare.net/hustwj/word-embeddings-what-how-and-whither for an excellent, if somewhat critical, survey. There are a number of useful tutorials such as http://www-personal.umich.edu/~ronxin/pdf/w2vexp.pdf. To use word2vec in NLTK, see http://www.johnwittenauer.net/language-exploration-using-vector-space-models.

Word2vec is not the first,[2] last or best[3] to discuss vector spaces, embeddings, analogies, similarity metrics, etc. But word2vec is simple and accessible. Anyone can download the code[4] and use it in their next paper. Any many do (for better and for worse).

Word2vec often takes on a relatively minor supporting role in these papers, largely bridging the gap between ascii input and an input format that is more appropriate for neural nets; word2vec is not particularly central to the main points of such papers, but nevertheless, in aggregate, the impact of word2vec is 'huge' (as Trump likes to say).

The importance of supporting roles should not be overlooked. Supporting roles could well have more impact in aggregate than leading roles. I use the term, supporting role, to include datasets and tools, as well as secondary sources: text books, surveys, discussions of discussions such as this, online courses, videos, etc. Successful supporting roles are richly rewarded with massive citations. Mitch Marcus and Steven Bird, former presidents of the ACL, have both made more than their share of technical contributions to the field, but their top citations, by far, are for supporting roles, datasets and tools such as the Penn Treebank and NLTK, respectively.

Supporting roles are not the first, last or best treatment of a topic, but they are often the most accessible (and the most popular on Google). The rich get richer...

2 Promising (if not convincing) initial successes

So you've written some code and uploaded it to github. Now, you are hoping the community will download it and cite it, but people aren't going to put in even the minimal effort required to try it out without motivation to do so. As mentioned in my last column, the hook doesn't need to be convincing. Promising is sufficient, and actually, promising might be better than convincing. If the hook is too convincing, the community won't even attempt to contribute improvements.

In word2vec's case, the hook is the analogy: *man* is to *woman* as *king* is to x. It is impressive that one can just download word2vec and discover that x is *queen*. Word2vec solves analogy tasks like this by trying all words, x', in the vocabulary, V, and finding the word that maximizes equation (1).

$$\hat{x} = ARGMAX_{x' \in V}\ sim(x', king + woman - man) \tag{1}$$

[2] There is considerable prior work, of course. The word2vec papers (Mikolov 2013b; Mikolov 2013a; Mikolov 2013c) cite relatively few papers before 2000, with the exception of Elman (1990) and Harris (1954). The discussion on word2vec mentions quite a few more on various topics such as distributional semantics (Weaver 1955; Firth 1957), vector spaces (Salton 1975), singular value decomposition (SVD)(Deerwester 1990), embeddings (Pereira 1993), PMI (pointwise mutual information) (Church 1990) and similarity estimates (Resnik 1995; Lin 1998).

[3] See https://www.aclweb.org/aclwiki/index.php?title=SAT_Analogy_Questions_(State_of_the_art) for rankings on an analogy task, as well as links to rankings on a number of similar tasks.

[4] https://github.com/dav/word2vec

https://doi.org/10.1017/S1351324916000334 Published online by Cambridge University Press

Words (e.g., *king*) are represented as vectors (e.g., *vec*(*king*)), sequences of K floats, where K is the number of internal dimensions, typically $K = 300$. Similarity is defined as a cosine

$$sim(a, b) \equiv cos(vec(a), vec(b)) = \frac{vec(a) \cdot vec(b)}{|vec(a)| \cdot |vec(b)|} \tag{2}$$

Addition and subtraction of words is expressed in terms of vector addition and vector subtraction. That is, $vec(king + woman - man) = vec(king) + vec(woman) - vec(man)$.

Levy (2014a) suggested that equation (1) can be reformulated in terms of three similarities, as in equation (3). In fact, they prefer equation (4), with small gains reported in Linzen (2016). We will return to this point in section 3.

$$\hat{x} = ARGMAX_{x' \in V}\ sim(x', king) + sim(x', woman) - sim(x', man) \tag{3}$$

$$\hat{x} = ARGMAX_{x' \in V}\ \frac{sim(x', king) \cdot sim(x', woman)}{sim(x', man)} \tag{4}$$

There are $\binom{4}{2} = 6$ possible pairwise (symmetric) similarities. I have found it useful to group the six similarities into three types of similarities: *vert*, *hor* and *diag*. Three of the similarities depend on x and three don't (labeled $\overline{x}$). There may be opportunities in some cases to infer x similarities from $\overline{x}$ similarities, especially when the same words show up multiple times on the test, but in different positions.

similarity type	$\overline{x}$	x
vert	$sim(man, woman)$	$sim(king, x)$
hor	$sim(man, king)$	$sim(woman, x)$
diag	$sim(woman, king)$	$sim(man, x)$

The intuition for these names comes from expressing the analogy as

$$\frac{man}{woman} = \frac{king}{queen} \tag{5}$$

While it is pretty amazing that such a simple method does as well as it does, the results are far from too successful, as illustrated in Table 1. Note that just two of the top ten candidates have the correct gender and number (*f, sg*).[5] Clearly, there is more work to be done, and plenty of opportunities for the next generation to make improvements.

How well does word2vec do? Table 2 reports accuracies on a range of analogy tasks using GoogleNews vectors.[6] Word2vec works better on some types of analogies than others. Performance is much better on question-words,[7] the standard test set distributed with the word2vec code, than on real SAT questions.[8]

[5] Bolukbasi (2016) use word2vec to study gender bias in documents. There is plenty of evidence of bias, though there is an opportunity to publish an evaluation of word2vec's effectiveness in detecting gender (and bias).

[6] https://github.com/mmihaltz/word2vec-GoogleNews-vectors

[7] https://github.com/arfon/word2vec/blob/master/questions-words.txt

[8] See https://www.aclweb.org/aclwiki/index.php?title=SAT_Analogy_Questions_(State_of_the_art) for information on obtaining the 374 SAT questions. I'm using a subset of the

https://doi.org/10.1017/S1351324916000334 Published online by Cambridge University Press

Table 1. *Top ten choices for* **x** *in man is to woman as king is to* **x**. *Although the top candidate, queen, is an impressive choice, many of the other top ten candidates are less impressive, especially the eight of ten candidates with incorrect gender/number. Candidates with larger hor similarities are more likely to inherit the desired gender and number features from woman. The overall score is close to hor + vert − diag, but not exactly because vector length normalization doesn't distribute over vector addition and subtraction*

Score	hor	vert	diag	x	Gender	Number
0.71	0.32	0.65	0.17	Queen	f	sg
0.62	0.25	0.64	0.19	Monarch	m	sg
0.59	0.40	0.52	0.25	Princess	f	sg
0.55	0.21	0.62	0.21	Crown˙prince	m	sg
0.54	0.27	0.62	0.28	Prince	m	sg
0.52	0.06	0.71	0.18	Kings	m	pl
0.52	0.26	0.45	0.12	Queen˙Consort	m	sg
0.52	0.21	0.47	0.10	Queens	f	pl
0.51	0.16	0.59	0.17	Sultan	m	sg
0.51	0.15	0.49	0.07	Monarchy	m	sg

Table 2. *Some types of analogies are easier than others, as indicated by accuracies for top choice (A_1), as well as top 2 (A_2), top 10 (A_{10}) and top 20 (A_{20}). The rows are sorted by A_1. These analogies and the type classification come from the questions-words test set, except for the last row, SAT questions. SAT questions are harder than questions-words*

A_1	A_2	A_{10}	A_{20}	N	Analogy type	Example
0.91	0.95	0.98	0.99	1,332	Comparative	$\frac{young}{younger} = \frac{wide}{wider}$
0.90	0.94	0.97	0.98	1,599	Nationality-adjective	$\frac{Ukraine}{Ukrainian} = \frac{Switzerland}{Swiss}$
0.90	0.93	0.97	0.98	1,332	Plural	$\frac{woman}{women} = \frac{snake}{snakes}$
0.87	0.94	1.00	1.00	1,122	Superlative	$\frac{young}{youngest} = \frac{wide}{widest}$
0.85	0.90	0.97	1.00	506	Family	$\frac{uncle}{aunt} = \frac{stepson}{stepdaughter}$
0.83	0.89	0.97	0.98	335	Capital-countries	$\frac{Tokyo}{Japan} = \frac{Tehran}{Iran}$
0.79	0.86	0.94	0.96	4,695	Capital-world	$\frac{Zagreb}{Croatia} = \frac{Dublin}{Ireland}$
0.78	0.84	0.98	0.99	1,056	Present-participle	$\frac{write}{writing} = \frac{walk}{walking}$
0.71	0.79	0.90	0.92	2,467	City-in-state	$\frac{Worcester}{Massachusetts} = \frac{Cincinnati}{Ohio}$
0.68	0.78	0.93	0.95	870	Plural-verbs	$\frac{write}{writes} = \frac{work}{works}$
0.66	0.82	0.97	0.98	1,560	Past-tense	$\frac{writing}{wrote} = \frac{walking}{walked}$
0.43	0.48	0.64	0.69	812	Opposite	$\frac{tasteful}{distasteful} = \frac{sure}{unsure}$
0.35	0.42	0.57	0.62	866	Currency	$\frac{Vietnam}{dong} = \frac{USA}{dollar}$
0.29	0.37	0.63	0.73	992	Adjective-to-adverb	$\frac{usual}{usually} = \frac{unfortunate}{unfortunately}$
0.01	0.02	0.08	0.10	190	SAT questions	$\frac{audacious}{boldness} = \frac{sanctimonious}{hypocrisy}$

https://doi.org/10.1017/S1351324916000334 Published online by Cambridge University Press

Table 3. *The two test sets have very different Venn diagrams. The 178 means that SAT has 178 words that are in a, but not b, c or d. The 14 means that there are 14 words in the overlap between b and d (and not a and c). SAT is more like what I was expecting with small overlaps. Since the vocabulary is much larger than the test set, it is unlikely to find the same word in multiple positions*

SAT questions					Questions-words				
#words	a	b	c	d	#words	a	b	c	d
178	1	0	0	0	431	0	1	0	1
175	0	0	1	0	399	1	0	1	0
160	0	1	0	0	75	1	1	1	1
156	0	0	0	1					
14	0	1	0	1					
4	1	0	1	0					
4	0	0	1	1					
3	1	1	0	0					
3	1	0	0	1					
2	1	1	0	1					
2	0	1	1	1					
2	0	1	1	0					

3 Error analysis and gaming the test

Given how different questions-words is from real SAT questions, I am concerned that questions-words has become a standard test set in the literature, as observed in Linzen (2016). Linzen (2016) then uses this test set to compare equations (3) and (4), and reports that latter is slightly better than the former. While that may well be the case, we need to make sure that such findings can be replicated over more than one test set, especially given the concern in Table 3.[9]

The task is to predict the last word, d, from the first three: a, b, c. Since the vocabulary ($|V| = 300,000$) is much larger than the test set (19,544 four-tuples), it should be unlikely to find the same word in two or more positions. Table 3 shows that this is the case for SAT, but not for questions-words. In fact, every word in questions-words appears in at least two positions. That could not happen by chance.

Once we knew that the test could be gamed, it was pretty straightforward to find a solution. Word2vec doesn't need to search over $|V| = 300,000$ words since no new words show up in the last position that don't also appear in other positions. Thus,

first 190 questions, labeled *190 FROM REAL SATs*; the rest have different attributions. Since the SAT analogy task is slightly different from the word2vec analogy task, I modified the SAT questions to be more comparable to the word2vec questions. The SAT questions give the student a pair of words (e.g., *audacious* and *boldness*), plus five more pairs. The task is to pick the best of the five pairs to complete the analogy. In this case, the correct pair is *sanctimonious* and *hypocrisy*. To make this task more comparable to the word2vec analogy task, I replaced the six pairs, with four words: *audacious*, *boldness*, *sanctimonious* and *hypocrisy*. The task is to predict the last word from the first three.

[9] Levy (2015) report results over a number of test sets. Their main point is the importance of hyperparameters, but they also find large differences by test set, casting doubt on claims that word2vec is an improvement over previous work such as PMI and SVD.

https://doi.org/10.1017/S1351324916000334 Published online by Cambridge University Press

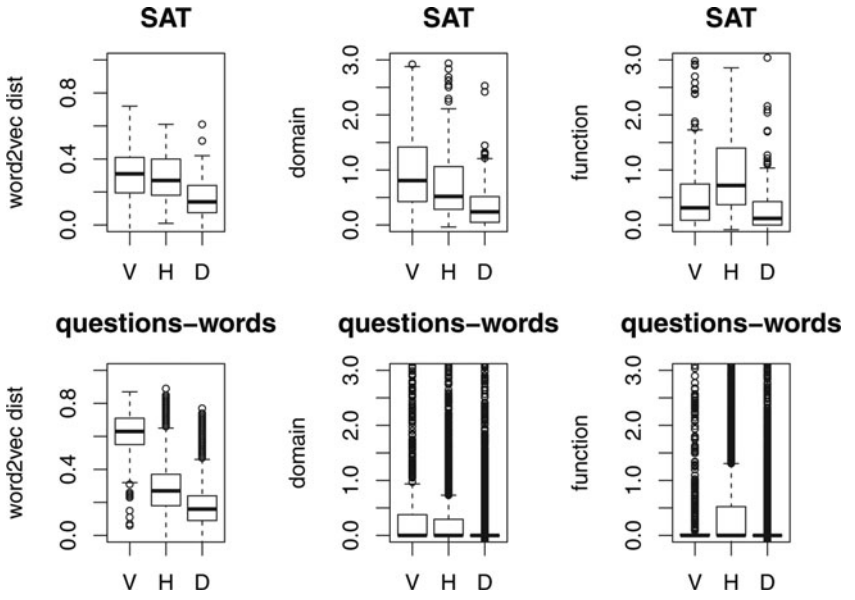

Fig. 1. Six boxplots comparing Vert, Hor and Diag. The three columns use three measures of similarity: (1) word2vec distance, (2) domain space and (3) function space. The top row uses SAT questions, and the bottom row uses questions-words. This plot is based on just $\overline{x}$ similarities, though the plot would not change much if we replaced $\overline{x}$ similarities with x similarities.

we can search over just $905 = 431 + 399 + 75$ words, rather than 300,000. Actually, one can cut down the search space even more by realizing that the mapping from *a* to *b* is the same as the mapping from *c* to *d*. That is, if one knows that *a* is *Argentina*, then *b* is either *Argentinean* or *peso*. And similarly, the same relationship holds between *c* and *d*. That is, if we know that *c* is *Argentina*, then *d* is either *Argentinean* or *peso*.

It turns out that this mapping is extremely constrained. *d* is usually (85 per cent) uniquely determined by *c*. Even when *d* is ambiguous (as in the case of *Argentinean* or *peso*), it isn't very ambiguous. If we know *c*, we can almost always (99.6 per cent) limit *d* down to one or two choices. Given these observations, it isn't surprising that we could come up with a 'cheating' solution that performs incredibly well on the test. My best solution achieved A_1 accuracy of 98.7 per cent.

Obviously, it is rather pointless to game the test, but the fact that it can be done should cast doubt on conclusions in the literature that are based solely on this (questionable) test set. Going forward, it is important to replicate results across a few test sets as in Figure 1. At first, I was looking at just one measure and just one test set (the bottom left plot in Figure 1). Based on that, I jumped to the conclusion that word2vec is stronger for vert (*a* versus *b*) than hor (*a* versus *c*). But the difference between vert and hor is largely gone in the upper left plot, suggesting that my conclusion was premature; the apparent difference between vert and hor was merely an artifact of the 'unusual' properties of the questions-words test set. To

https://doi.org/10.1017/S1351324916000334 Published online by Cambridge University Press

support generalizations to a population of interest, it is crucial that the test set be a random sample of that population.

Figure 1 compares word2vec distances with two additional similarity measures proposed in Turney (2012). All of three similarity measures are related to PMI (Church 1990) in interesting ways,[10] but domain and function were designed to complement one another. Both follow Firth's advice, *you shall know a word by the company it keeps*, but domain does so by looking at nouns in nearby contexts, and function does so by looking at verbs in nearby contexts. The last two plots on the top row show that the difference matters, at least on the SAT test set. Note that domain and function distinguish vert and hor on the top row (but the bottom row is less conclusive, probably because of flaws in questions-words test set).

4 Conclusions

Word2vec has racked up plenty of citations because it satisfies both of Kuhn's conditions (Kuhn 2012) for emerging trends: (1) a few initial (promising, if not convincing) successes that motivate early adopters (students) to do more, as well as (2) leaving plenty of room for early adopters to contribute and benefit by doing so. Perhaps, it is a bit of an overstatement to compare word2vec to Kuhn's scientfic revolutions, but nevertheless, word2vec has had a huge impact on the field. Word2vec is playing an important supporting role. Anyone can download the code and use it in their next paper. Any many do (for better and for worse). The most cited paper is often not the first, or the last, or even the best. Simplicity and accessibility are preferred over timing and accuracy/correctness. The community needs to be careful, however, not to be too convinced by initial promising results. In particular, we need to replicate results over more credible test sets before jumping to premature conclusions.

References

Bolukbasi, T., Chang, K.-W., Zou, J., Venkatesh Saligrama Adam and Kalai 2016. Man is to computer programmer as woman is to homemaker? Quantifying and Reducing Stereotypes in Word Embeddings. To appear in Advances in Neural Information Processing Systems (NIPS).

Church, K., and Hanks, P. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics* **16**(1): 22–29.

Deerwester, S., Dumais, S., Furnas, G., Landauer T., and Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* **41**(6): 391–407.

Elman, J. 1990. Finding structure in time. *Cognitive Science* **14**(2): 179–211.

Eysenbach, G. 2006. Citation advantage of open access articles. *PLoS Biol* **4**(5): e157. doi:10.1371/journal.pbio.0040157

Firth, J. 1957. *A Synopsis of Linguistic Theory, 1930–1955*. Oxford: Blackwell.

Harris, Z. 1954. Distributional structure. *Word* **10**(2–3): 146–162.

[10] See Levy (2014b) for connections between word2vec distances and PMI.

Kuhn, T. 2012. *The Structure of Scientific Revolutions*. Chicago, USA: University of Chicago Press.
Lawrence, S. 2001. Free online availability substantially increases a paper's impact. *Nature* **411**(6837): 521–521.
Levy, O., and Goldberg, Y. 2014a. Linguistic regularities in sparse and explicit word representations. *CoNLL* 171–180.
Levy, O., and Goldberg, Y. 2014b. Neural word embedding as implicit matrix factorization *NIPS* 2177–2185.
Levy, O., and Goldberg, Y. 2014c. Word2vec explained: deriving Mikolov et al.'s negative-sampling word-embedding method. arXiv:1402.3722
Levy, O., Goldberg, Y., and Dagan, I. 2015. Improving distributional similarity with lessons learned from word embeddings. *TACL* **3**: 211–225.
Lin, D. 1998. An information-theoretic definition of similarity. *ICML* **98**: 296–304.
Linzen, T. 2016. Issues in evaluating semantic spaces using word analogies. *arXiv:1606.07736*.
Mikolov, T., Chen, K., Corrado, G., and Dean, J. 2013a. Efficient estimation of word representations in vector space. ICLR. https://arxiv.org/abs/1301.3781.
Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. 2013b. Distributed representations of words and phrases and their compositionality. *NIPS* 3111–3119.
Mikolov, T., Yih, W.-T., and Zweig, G. 2013c. Linguistic regularities in continuous space word representations. *HLT-NAACL* **13**: 746–751.
Pereira, F., Tishby, N., and Lee, L. 1993. Distributional clustering of English words. *ACL* 183–190.
Resnik, P. 1995. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *J. Artif. Intell. Res. (JAIR)* **11**: 95–130.
Salton, G., Wong, A., and Yang, C.-S. 1975. A vector space model for automatic indexing. *CACM* **18**(11): 613–620.
Turney, P. 2012. Domain and function: a dual-space model of semantic relations and compositions. *Journal of Artificial Intelligence Research* **44**: 533–585.
Weaver, W. 1955. Translation. *Machine Translation of Languages*, vol. 14. Cambridge: Technology Press, MIT, pp. 15–23.