

Can word vectors help corpus linguists?

Guillaume Desagulier

▶ **To cite this version:**

Guillaume Desagulier. Can word vectors help corpus linguists?. 2017. halshs-01657591v1

HAL Id: halshs-01657591

https://shs.hal.science/halshs-01657591v1

Preprint submitted on 6 Dec 2017 (v1), last revised 3 Oct 2018 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Can word vectors help corpus linguists?

Guillaume Desagulier[1]

[1]MoDyCo — Université Paris 8, CNRS, Université Paris Nanterre, Institut Universitaire de France

Abstract

Two recent methods based on distributional semantic models (DSMs) have proved very successful in learning high-quality vector representations of words from large corpora: `word2vec` (Mikolov, Chen, et al. 2013; Mikolov, Yih, et al. 2013) and `GloVe` (Pennington et al. 2014). Once trained on a very large corpus, these algorithms produce distributed representations for words in the form of vectors. DSMs based on deep learning and neural networks have proved efficient in representing the meaning of individual words. In this paper, I assess to what extent state-of-the-art word-vector semantics can help corpus linguists annotate large datasets for semantic classes. Although word vectors suggest decisive opportunities for resolving semantic annotation issues, it has yet to improve in terms of its representation of polysemy, homonymy, and multiword expressions.

Keywords: corpus linguistics, word vectors, distributional semantic models, semantic annotation

1 Introduction

Annotating linguistic data for semantic categories is arguably one of the last fronteers in corpus linguistics. Divjak and Fieller (2014, 208) make it sound as if manual annotation were a defining feature of traditional corpus linguistics: "A common myth is that corpus linguists do everything automatically, which would make corpus linguistic techniques unsuited for the study of meaning." Yet, with the increasing availability of very large corpora, such as those compiled from the web (Baroni, Bernardini, Ferraresi, & Zanchetta 2009), corpus linguists may have to handle very large datasets in the near future. Contemporary semanticists commonly extract corpus data and annotate the datasets in a manual, a priori fashion. If the dataset is too large, manual annotation may become unfeasible.

No principled opposition exists against automatization in computational linguistics, an area that has witnessed dramatic progress in distributional semantics. Based on the distributional hypothesis (Harris 1954; Miller & Charles 1991), according to which words that appear in similar contexts have similar meanings, distributional semantic modeling contends that it is possible to approximate what humans do when they learn word meanings via similarity judgments. Distributional semantic models approximate this knowledge by operationalizing semantic similarities between linguistic units based on their distributional properties in large bodies of natural language data. Specifically, once trained on a very large corpus, distributional semantic models produce distributed representations for words in the form of strings of real numbers called vectors. Words with similar vector representations have similar meanings.

Quite recently, two state-of-the-art unsupervised learning methods claim to have taken word-vector representations to the next level: `word2vec` (Mikolov, Chen, Corrado, & Dean 2013a; Mikolov, Sutskever, Chen, Corrado, & Dean 2013c; Mikolov, Yih, & Zweig 2013d) and `GloVe` (Pennington, Socher, & Manning 2014). Based on neural networks, these models have met with immediate success even though there is ongoing debate as to whether the quality of the models has been assessed properly. The enthusiasm in the computational linguistics community comes from the good performance of `word2vec` and `GloVe` in downstream tasks (mapping linear relationships, assessing similarity, etc.).

Starting from the assumption that manual annotation is the gold standard, this paper assesses to what extent state-of-the-art word-vector semantics can help the corpus linguist annotate words for semantic categories. Section 2 addresses common issues with semantic annotation and the need for automatization in this task. Section 3 examines the relevance of distributional semantic modeling in this respect. Section 4 shows how `GloVe` performs in an annotation task involving adjectives. Section 5 discusses the limits of word vectors. In Section 6, I conclude that although the computational performance of word-vector semantics is indeed noteworthy, this technique has yet to improve before corpus linguists can fully benefit from it.

2 Common issues with semantic annotation

Semantic annotation is a common task in corpus linguistics. In Desagulier (2015), I compared the adverbs *quite* and *rather* in their preadjectival and predeterminer uses, as exemplified below.

(1) a. That has proved to be *a quite difficult* question to answer. (preadjectival)
b. That has proved to be *quite a difficult* question to answer. (predeterminer)

a. That is *a rather difficult* question to answer. (preadjectival)
b. That is *rather a difficult* question to answer. (predeterminer)

To remain within the strict limits of the preadjectival vs. predeterminer alternation, I extracted only those patterns that involved both alternants, namely <*a(n) quite/rathe*r ADJ NP> and <*quite/rather a(n)* ADJ NP>, from the `British National Corpus` (XML Edition), henceforth `BNC-XML`. I gathered all occurrences in a dataset, of which Table 1 is a sample. The full dataset consists of 3086 observations. Five variables are displayed:

- the name of the corpus file,
- the construction (predeterminer vs. preadjectival),
- text information (W fict prose, W ac:humanities arts, S conv, etc.),
- the exact match (i.e. the adverb in context),
- the intensified adjective.

To see if meaning has any bearing on the preadjectival vs. predeterminer alternation, the adjectives must be annotated semantically. In corpus linguistics, the most widespread approach consists in annotating examples manually. The adjectives in Table 1 were therefore manually annotated using a tagset inspired by Dixon and Aikhenvald (2004). Classes that matched the data better were added in an *ad hoc* fashion. The semantic tags were appended to the original dataset, as shown in Table 2.

Table 1: a sample data frame compiled from the BNC-XML

corpus_file	construction	text info	match	adjective
K1J.xml	predeterminer	W news script	*quite a hot shot*	hot
G2W.xml	preadjectival	W pop lore	*a rather hot seller*	hot
KRT.xml	preadjectival	S brdcast news	*a quite clear position*	clear
J0V.xml	predeterminer	W ac:humanities arts	*quite a clear understanding*	clear
CHE.xml	predeterminer	W biography	*quite a clear view*	clear
FEV.xml	predeterminer	W nonAc: nat science	*quite a clear picture*	clear
EWR.xml	preadjectival	W nonAc: polit law edu	*a quite clear line*	clear
CRK.xml	predeterminer	W religion	*quite a clear stand*	clear
HA7.xml	preadjectival	W fict prose	*a rather clouded issue*	clouded
KPV.xml	predeterminer	S conv	*quite a cold day*	cold
G3B.xml	preadjectival	W biography	*a rather cold morning*	cold
AB5.xml	preadjectival	W biography	*a rather cold person*	cold
CDB.xml	predeterminer	W fict prose	*rather a cold note*	cold
K23.xml	preadjectival	W news script	*a rather colder winter*	colder

Table 2: a sample data frame with manual annotation

corpus_file	construction	text info	match	adjective	semantic class
K1J.xml	predeterminer	W news script	*quite a hot shot*	hot	value_positive
G2W.xml	preadjectival	W pop lore	*a rather hot seller*	hot	value_positive
KRT.xml	preadjectival	S brdcast news	*a quite clear position*	clear	clearness
J0V.xml	predeterminer	W ac:humanities arts	*quite a clear understanding*	clear	clearness
CHE.xml	predeterminer	W biography	*quite a clear view*	clear	clearness
FEV.xml	predeterminer	W nonAc: nat science	*quite a clear picture*	clear	clearness
EWR.xml	preadjectival	W nonAc: polit law edu	*a quite clear line*	clear	clearness
CRK.xml	predeterminer	W religion	*quite a clear stand*	clear	clearness
HA7.xml	preadjectival	W fict prose	*a rather clouded issue*	clouded	unclearness
KPV.xml	predeterminer	S conv	*quite a cold day*	cold	temperature_cold
G3B.xml	preadjectival	W biography	*a rather cold morning*	cold	temperature_cold
AB5.xml	preadjectival	W biography	*a rather cold person*	cold	value_undesirable
CDB.xml	predeterminer	W fict prose	*rather a cold note*	cold	psych_stim_bad
K23.xml	preadjectival	W news script	*a rather colder winter*	colder	temperature_cold

The main asset of manual annotation is that polysemous items are assigned one meaning per context. For example, *hot* in *quite a hot shot* is assigned the tag `value_positive` because it is used figuratively. The adjective qualifies a noun whose referent is talented or successful. The same adjective would be assigned the tag `temperature_hot` if it were used literally in *quite a hot summer*. The same human expertise distinguishes *clouded* in contexts where it means 'unclear' from contexts where it means 'covered by clouds'.

To reduce the bias of a single annotator, the same table can be annotated by several researchers on the basis of a tagset that has been agreed upon beforehand. The annotation is then verified by means of Cohen's κ (Cohen 1960, 1968), which measures the degree of agreement between several annotators.[1]

One major limitation of manual annotation is that once a coding scheme has been chosen, it cannot be amended anymore. The annotator must therefore know the data perfectly before determining what tags should be used. Because of this, human annotation might be considered the gold standard.

However, manual annotation comes with limitations. First, it is a time-consuming and energy-intensive process. Therefore, the size of the dataset must be reasonable. This problem is all the more acute as large corpora are becoming the norm. When it was first published in the mid-1990s, the 100M-word `BNC` was considered a very large corpus (Burnard 2000). Compared to today's 2.25B-word `ukWac` corpus of English or Sketch Engine's 13B-word `enTenTen12` corpus, the `BNC-XML` is not so large anymore. Thanks to the increased availability of digital texts and text-editing tools, we are currently witnessing an arms race in corpus size.

On the one hand, very large corpora are a good thing since they allow the linguist to investigate rare linguistic forms. One field in which size is critical is the study of productivity. As described by Baayen (2001), productivity measures rely on the idea that the number of hapax legomena of a given grammatical category correlates with the number of neologisms in that category, which in turn correlates with the productivity of the rule at work. Thus, lexical productivity is a factor of both a large number of low-frequency forms and a low number of high-frequency forms. In a productivity study, the corpus must be large enough to yield a minimal number of rare forms. Zeldes (2012) recommends that samples be obtained from very large corpora such as those compiled from the web (Baroni et al. 2009).

On the other hand, corpus linguists are well aware that very large corpora are difficult to handle. Any collection of texts generates some noise, i.e. unwanted data. This phenomenon is captured by the precision vs. recall trade-off. The precision of a corpus query is the proportion of relevant hits and the total number of returned occurrences. Recall is the proportion of relevant retrieved hits with respect to the total number of relevant hits in the corpus. Corpus linguists tend to maximize recall so as to minimize silence (the unhappy situation when the query returns zero hit), which undermines precision. When precision is not optimal, the linguist has to filter the output manually. The larger the corpus and the broader the query, the larger the dataset and the more tedious the clean up. If the dataset is too large, its manual annotation becomes infeasible.

For this reason, not all corpus linguists are willing to embrace the age of "big data". They rightly argue that the most important quality of a corpus is not its size but its sampling scheme. A good sampling scheme guarantees that the corpus is as representative and balanced as possible. A corpus is representative when it is faithful to the variability that characterizes the target

[1] For each type of tag, the score ranges from 0 to 1. A score of 0.4 or below is generally considered unsatisfactory, whereas a score of 0.6 or above is considered satisfactory. For more details on how Cohen's κ is used in the field of cognitive semantics, see Glynn (2010, 250). In Glynn's study, the verb *bother* is annotated by means of four categories: affect, theme, agent, and cause.

Table 3: a sample data frame with USAS annotation

corpus_file	construction	text info	match	adjective	USAS tag	sem class
K1J.xml	predeterminer	W news script	*quite a hot shot*	hot	O4.6+	Temperature_Hot_on_fire
G2W.xml	preadjectival	W pop lore	*a rather hot seller*	hot	O4.6+	Temperature_Hot_on_fire
KRT.xml	preadjectival	S brdcast news	*a quite clear position*	clear	A7+	Likely
J0V.xml	predeterminer	W ac:humanities arts	*quite a clear understanding*	clear	A7+	Likely
CHE.xml	predeterminer	W biography	*quite a clear view*	clear	A7+	Likely
FEV.xml	predeterminer	W nonAc: nat science	*quite a clear picture*	clear	A7+	Likely
EWR.xml	preadjectival	W nonAc: polit law edu	*a quite clear line*	clear	A7+	Likely
CRK.xml	predeterminer	W religion	*quite a clear stand*	clear	A7+	Likely
HA7.xml	preadjectival	W fict prose	*a rather clouded issue*	clouded	O4.3	Colour_and_colour_patterns
KPV.xml	predeterminer	S conv	*quite a cold day*	cold	O4.6-	Temperature_Cold
G3B.xml	preadjectival	W biography	*a rather cold morning*	cold	B2-	Disease
AB5.xml	preadjectival	W biography	*a rather cold person*	cold	O4.6-	Temperature_Cold
CDB.xml	predeterminer	W fict prose	*rather a cold note*	cold	O4.6-	Temperature_Cold
K23.xml	preadjectival	W news script	*a rather colder winter*	colder	O4.6–	Temperature_Cold

language. It is balanced when the proportion of the sampled elements that make it representative corresponds to the proportion of the same elements in the target language. Depending on the case study and the language investigated, using a smaller corpus is not a bad thing. For example, Hollmann and Siewierska (2007), and Boas and Schuchard (2012) show that, once compiled into a corpus, scarce resources can go a long way in the study of minority languages or dialects.

Those corpus linguists who cannot resist the appeal of very large corpora must find a way to handle datasets that are so large that manual cleaning becomes hardly feasible. One option consists in using a semantic tagger. In Table 3, the adjectives were tagged with the UCREL Semantic Analysis System (USAS), as described by Piao, Bianchi, Dayrell, D'Egidio, and Rayson (2015).

The list of adjectives in the dataset is matched against a dictionary where words are preassigned a specific tag. Each tag corresponds to a semantic category. The whole tagset is available at http://ucrel.lancs.ac.uk/usas/. This approach is both fast and automatic. Its main downside is that the tags do not take the specific context of the occurrence into account. Polysemous items are therefore assigned one general meaning, which is often at odds with the contextual specificities of the occurrence. This problem is visible in Table 3, where *hot* does not imply anything close to temperature or fire in *quite a hot shot*, or where *clouded* has little to do with color and color patterns. Note also that *cold* in *a rather cold morning* has been mistaken for a noun and has been assigned the tag Disease incorrectly. Paradoxically, the price to pay for automatization is manual inspection and correction.

Ideally, semantic annotation should be indexed on context. Because meaning cannot be accessed in a straightforward fashion, it is generally inferred from word distributions. This is where distributional semantic models have a part to play.

3 DSMs

The idea that the meaning of a word can be inferred from its distribution is not new in linguistics. According to the distributional hypothesis, semantically similar words tend to have similar contextual distributions (Harris 1954; Miller & Charles 1991). Underlying the distributional hypothesis is the idea that 'you shall know a word by the company it keeps" (Firth 1957, 179). The distributional hypothesis relies on a structuralist conception of meaning similarity. What

it aims at is the meaning derived from positions in a text, not the kind of psycholinguistically realistic meaning that contemporary cognitive semantics is after. Compare (2) and (3):

(2) We had a wonderful time in Paris.

(3) We had a terrible time in Paris.

(4) We had a wonderful dinner in Paris.

Wonderful and *terrible* have the same distribution but opposite meanings. Yet, these antonyms denote properties that belong to the broad field of appreciation (whether positive or negative). More problematic is the semantic relation that holds between *time* and *dinner* in (2) and (4), respectively. Critics of the hypothesis may argue that *time* and *dinner* have nothing in common. Defenders of the hypothesis will reply that they have similar meanings by virtue of their distributions: both belong to the broad class of nouns that allow for appreciative qualification. The distributional hypothesis may be based on a view of meaning that is too restrictive, it is nevertheless computationally viable.

Several implementations of the distributional hypothesis have been suggested at the level of computational modeling. These implementations are known as distributional semantic models (henceforth DSMs). Arguably, Latent Semantic Analysis (LSA) is one of the most popular DSMs (Deerwester, Dumais, Furnas, Landauer, & Harshman 1990; Schütze 1992). LSA analyzes the links between documents by looking at the terms that these documents have in common. It consists in building a matrix whose lines correspond to the terms and whose columns correspond to the documents. The cells specify the number of terms contained in each document. How important a term is to a document is weighted by means of the TF–IDF (*term-frequency – inverse document frequency*) statistic. The matrix is then transformed into a series of relations between terms and concepts, and finally into a series of relations between concepts and documents. The links between the documents depend on the statistical properties of the terms that they contain.

The principle underlying DSMs is simple: the meaning of a word is computed from the distribution of its co-occurring neighbors. The words are generally represented as vectors, i.e. numeric arrays that keep track of the contexts in which target terms appear in the large training corpus. The vectors are proxies for meaning representations. DSMs are not new to those linguists who are familiar with NLP (Padó & Lapata 2007). Vector space models of word co-occurrence have been applied to tasks such as synonymy detection, concept categorization, verb selectional preferences, argument alternations, etc. What is new is their dramatic improvements thanks to deep learning and neural networks.

Converting a corpus into lexical vectors involves three steps:

1. the corpus is split into words;

2. vector representations of words are learned;

3. a matrix of vectors is output.

Table 4 shows what a matrix of vectors looks like.[2] Each adjective is described by a vector consisting of 300 numbers (i.e. 300 dimensions). To some extent, the vector can be considered as the digital fingerprint of a word type. If two words have similar vectors, this is because they have similar distributions. If we follow the distributional hypothesis, this means that the words have similar meanings. The quality of the vector representation is a function of the number

[2]It is sampled from a matrix of vectors obtained with `GloVe` (see below) on the basis of the `Common Crawl` dataset.

of dimensions. Typically, these range from 50 to 1000. The larger the number of dimensions, the finer-grained the vector representation. A fine-grained vector representation means that the vector keeps track of the contexts where the word appears. Note, however, that a very large number of dimensions makes it harder to spot similarities between words.

Table 4: A data frame augmented with word vectors (8 dimensions out of a total of 300)

adjective	V1	V2	V3	V4	V5	V6	V7	V8	...
hot	-0.39125	-0.539250	-0.117350	0.0913670	0.227630	-0.369510	-2.5946	-0.30800	...
clear	-0.16065	0.019293	0.068165	0.0031946	0.127560	0.099395	-3.2602	0.67777	...
clouded	-0.29590	-0.454450	-0.509670	-0.7854500	-0.186140	0.440430	-1.0733	0.36048	...
cold	-0.51470	-0.280980	-0.251460	-0.3151800	0.057145	-0.237330	-3.0243	0.19498	...
colder	-0.22718	0.389940	-0.528260	-0.2897200	0.155540	0.245550	-1.7086	0.40310	...
...	...	...	...	...	...	...	...	...	...

Once a matrix of vectors is produced, the nearest neighbors to a given target word can be computed and projected on a three-dimensional plot. In Figure 1, the nearest neighbors of *clouded* (Figure 1a) and *cold* (Figure 1b) are selected on the basis of cosine similarity.[3] Each figure focuses on one portion of the semantic space around the target word. Because semantic spaces are high-dimensional, four steps are required to reduce the number of dimensions and make three-dimensional plotting possible:

1. the nearest *n* words to the target are computed based on vector similarities;
2. all cosine values between the nearest neighbors as well as the original input are computed and stored in a matrix;
3. the cosine matrix is submitted to a principal component analysis;
4. the dimensions of the resulting matrix are reduced to three, so that a three-dimensional vector is assigned to each of the *n* neighbors and the target word.

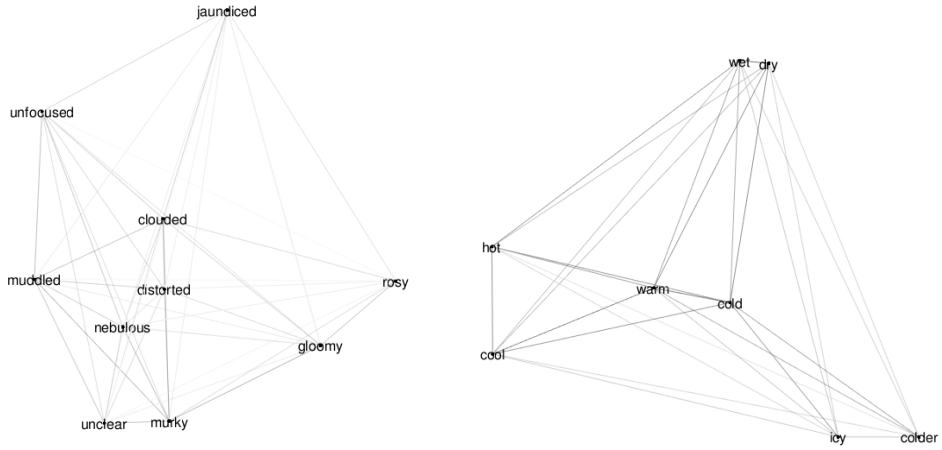
These vectors can thus be used to generate coordinates for 3D plotting.

Whereas traditional approaches, such as LSA, are based on counts because the vectors are indexed on co-occurrence counts, more recent approaches are based on prediction (Baroni, Dinu, & Kruszewski 2014). Recently, two predictive DSMs have proved very successful in learning high-quality vector representations of words from large corpora: `word2vec` (Mikolov et al. 2013a; Mikolov et al. 2013c; Mikolov et al. 2013d) and `GloVe` (Pennington et al. 2014). Based on neural networks, they learn word embeddings that capture the semantics of words by incorporating both local and global corpus context, and account for homonymy and polysemy by learning multiple embeddings per word. Once trained on a very large corpus, these models produce distributed representations for words: each word type is represented as a dense vector. They are predictive because they hinge on supervised context prediction training. In other words, the vector is designed so that it is good at predicting other words appearing in its context. The neighboring words are also represented by vectors.

The first method, `word2vec`, hinges on two model architectures: CBOW and continuous skip-gram (Rong 2014).[4] CBOW stands for 'Continuous Bag of Words'. It predicts a word

[3]The cosine similarity between two vectors is a measure that calculates the cosine of the angle between them in the vector space. Cosine similarity ranges from -1, in which case two words have opposite meanings, to 1, in which case two words have identical meanings. If cosine similarity is zero, this is because of a lack of semantic correlation.

[4]See also https://code.google.com/archive/p/word2vec/.

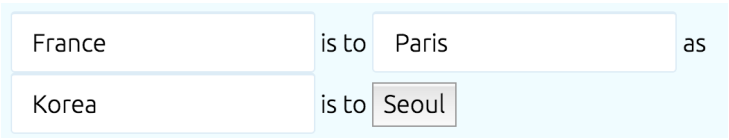


(a) The nearest neighbors of *clouded*

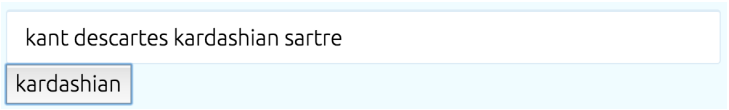
(b) The nearest neighbors of *cold*

Figure 1: Nearest neighbors plotted on the basis of word vectors

given its context. Continuous skip-gram does the reverse: it predicts a word context given the word itself. In the original project, `word2vec` was trained by Tomas Mikolov and his team at Google on the 100B-word *Google News* dataset. Because `word2vec` is an unsupervised task, the quality of the training cannot be evaluated objectively. Radim Řehůřek has developed a web application to evaluate the training manually (http://rare-technologies.com/word2vec-tutorial/#app). The application consists of several analogy tasks, two of which are illustrated in Figure 2.



(a) Inference based on linear analogy



(b) Finding the odd one out

Figure 2: Analogy tasks

`GloVe` hinges on global log-bilinear regression models and a weighted least squares model training on global word–word co-occurrence counts. According to Pennington et al. (2014) this combination of models improves upon `word2vec`:

> Methods like skip-gram may do better on the analogy task, but they poorly utilize the statistics of the corpus since they train on separate local context windows instead of on global co-occurrence counts.

`GloVe` is claimed to require fewer dimensions to achieve greater accuracy over `word2vec`, as shown in Table 5, adapted from Pennington et al. (2014).

On the basis of the above comparison, I chose to train `Glove` on two corpora:

- the 100M-word `BNC-XML` corpus,

Table 5: A comparison of how CBOW, Skip-gram, and GloVe perform on analogy tasks

model	dimensions	size in words	accuracy
CBOW	1000	6B	63.7%
Skip-gram	1000	6B	65.6%
GloVe	300	42B	75%

Table 6: How GloVe performs depending on corpus size and dimensionality

corpus	dimensions	semantic accuracy	syntactic accuracy	**total accuracy**
BNC-XML	50	18.37%	34.40%	**27.76%**
BNC-XML	300	39.63%	48.13%	**44.34%**
GloWbE	300	52.83%	43.08%	**47.47%**

- the 1.9B-word Corpus of Global Web-Based English, henceforth GloWbE (Davies 2013).

To see to what extent dimensionality has an impact on training accuracy, I compared the accuracy scores when the training involved 50 dimensions for the BNC-XML and 300 dimensions for the BNC-XML and GloWbE. Table 6 displays the results.

Because greater accuracy is obtained with GloWbE than with the BNC-XML when the number of dimensions is identical, the size of the corpus matters. However, the gain is not remarkable (44.34% $\rightarrow$ 47.47%). The gain in accuracy is much more decisive when the number of dimensions is raised from 50 to 300 for the BNC-XML (27.76% $\rightarrow$ 44.43%).

Admittedly, the BNC-XML and GloWbE are quite small when compared to the size of the training corpora that are normally used in deep learning, which reflects in the relatively low accuracy scores. Large corpora are preferred when NLP engineers want to obtain semantic spaces that approximate the semantic knowledge of native speakers. The enormous size of the training databases is therefore a way of avoiding distributional bias. If the training corpus is too small, the vector representations will be biased because of the limited range of topics and also because of idiosyncrasies in the corpus. The larger the corpus, the more likely the words will be used in a variety of semantic and syntactic contexts. To corpus linguists, avoiding bias is a pipe dream. The scores in Table 6 reflect vector quality only with respect to an ideal semantic space. Given that bias is both unavoidable and not a necessarily bad thing in corpus linguistics, we should not take these scores at face value. In other words, just because semantic and/or syntactic accuracy is far below the desired 100% does not mean that the word-vector training is useless.

Table 7 lists the cosine distance between the adjective *ironic* and its neighboring vectors in the BNC-XML and GloWbE. With respect to the BNC-XML, cosine distance is compared at two levels of dimensionality: 50 and 300 dimensions.

With respect to the BNC-XML, we see that cosine-distance scores are higher when they are computed on the basis of a matrix of shorter, denser vectors (50-dimensional word embeddings whose values are non-zero). Denser vectors, i.e. vectors whose values are mostly non-zero, are considered better at capturing general similarities between synonyms because the gist of the semantic relation is not diluted in a myriad of contexts. Conversely, sparser vectors, i.e. vectors whose values are mostly equal to zero, are expected to capture finer-grained similarities between synonyms but have the disadvantage that they can ignore obvious relations such as

Table 7: Cosine distance between *ironic* and its neighboring vectors in the BNC-XML and GloWbE

BNC-XML (50 dimensions)		BNC-XML (300 dimensions)		GloWbE (300 dimensions)	
word	cosine distance	word	cosine distance	word	cosine distance
curious	0.775490	*ironical*	0.537345	*surprising*	0.558550
remark	0.772809	*irony*	0.532534	*bizarre*	0.540507
gesture	0.766789	*remark*	0.517328	*somewhat*	0.537542
ironical	0.751805	*cynical*	0.515127	*twist*	0.528551
irony	0.759441	*wry*	0.508895	*embarrassing*	0.526689
absurd	0.734588	*humour*	0.490868	*irony*	0.520878
reassuring	0.731953	*somewhat*	0.472568	*absurd*	0.512750
dismissive	0.730330	*curious*	0.466679	*shocking*	0.511472
cynical	0.727852	*sad*	0.462698	*depressing*	0.495893
poignant	0.726002	*smile*	0.462698	*seems*	0.494366

between *couch* and *sofa*, or *car* and *automobile*. The difference does not show here because the vectors are dense, regardless of whether they are 50- or 300-dimensional. If we compare the 300-dimensional vectors of the BNC-XML and those of GloWbE, the cosine-distance scores are slightly higher for the latter because of the difference in corpus size.

Regardless of the number of dimensions and the size of the corpus, the results make sense. The vast majority of nearest neighbors consists of adjectives that can be considered near synonyms of *ironic*: *curious*, *surprising*, *bizarre*, The vectors capture the two main meanings of *ironic*, namely the psychological attitude (*cynical*, *wry*, *ironical*, *depressing*) and the feeling of incongruity between what happens and what was expected to happen (*curious*, *absurd*, *suprising*, *bizarre*, *embarrassing*). The results also include collocates of *ironic* such as nouns, verbs, and adverbs: *ironic* + NP (*remark*, *gesture*, *twist*, *humour*, *smile*), VP + *ironic* (*seem*), and ADV + *ironic* (*somewhat*). The presence of collocates comes from the distributional nature of vectors: the syntagmatic proximity between words yields similar representations. The noun *irony* has a special status in the list. It is not a collocate of *ironic* and cannot be considered a near synonym because it belongs to a distinct paradigmatic class. The noun and the adjective have similar vectors because they occur in similar contexts, i.e. contexts involving a large portion of identical collocates.

On the whole, the performance of GloVe is encouraging, even when the corpus is relatively small. Relatively consistent semantic clusters and satisfactorily rich lexical relations are obtained. In the section below, I implement word vectors as a proxy for semantic annotation.

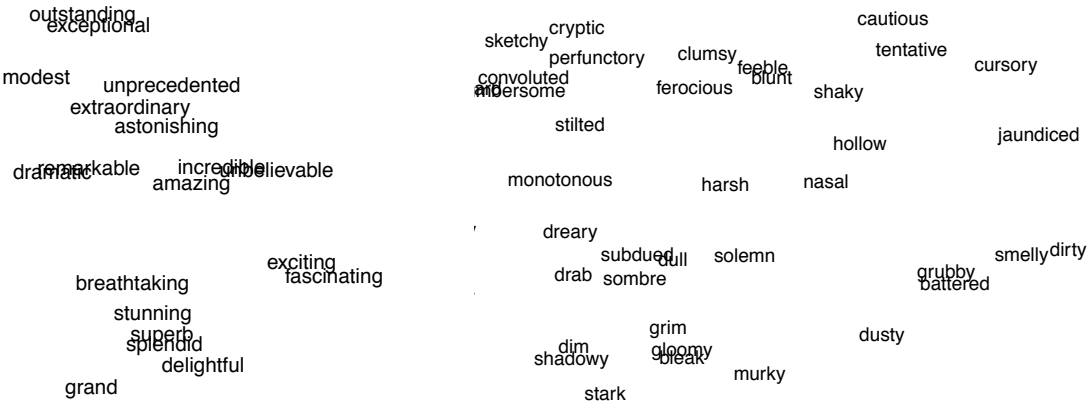
4 An exploratory annotation task

As mentioned above, word-vectors are commonly obtained from very large corpora. GloVe provides word-vector models that have been pre-trained on several datasets, the largest of which contains 42B word tokens, 1.9M word types and 300 dimensions.[5] The default word2vec model is trained on a dataset that contains 100B word tokens. This idea is based on the belief that if machine learning is performed on the largest possible corpus, bias ends up dissolving as all

[5] Common Crawl, http://nlp.stanford.edu/data/glove.42B.300d.zip.

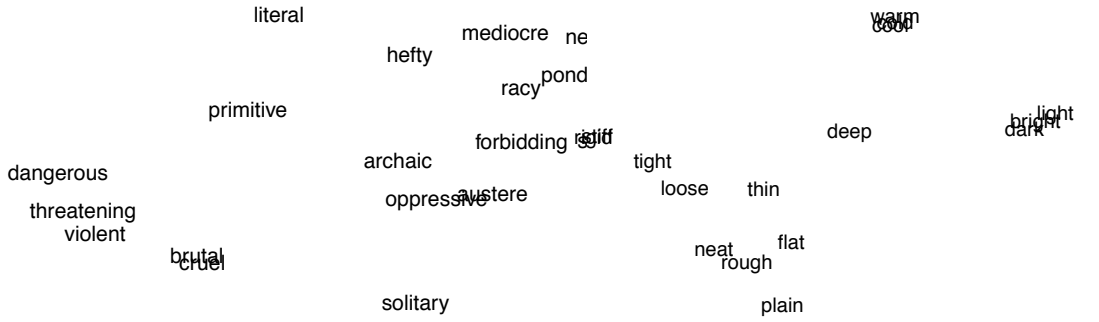
the syntactic and semantic possibilities are exhausted. This belief is largely shared in the NLP community. Following Zipf (1949), corpus linguists contend that even the largest collection of texts in a given language does not contain instances of all types in that language. In other words, bias is unavoidable. Most corpus linguists rightly claim that they do not aim to explain all of a language in every study and that the limits of their generalizations are the limitations of the corpus. Those limitations are interesting in themselves, and so is bias.

For the above reason, and also to benefit from the maximal cosine similarity observed in Table 7, I annotated the adjectives from Table 1 with 50-dimensional vectors obtained from the `BNC-XML`. The resulting table is high dimensional. To summarize it, it is necessary to reduce its dimensionality before we plot it on a two-dimensional plane. van der Maaten and Hinton (2008) have devised a dimensionality reduction technique known as t-Distributed Stochastic Neighbor Embedding (t-SNE). They claim that t-SNE dimensionality reduction is particularly well suited for the visualization of high-dimensional datasets. Figure A.1 in Section A is the graphic output of a Barnes-Hut implementation of t-SNE. Most adjectives cluster on the basis of the qualities that they denote, e.g. positive qualities (Figure 3a), negative qualities (Figure 3b), attitudes and psychological dispositions (Figure 3c), and physiological stimuli (Figure 3d). Because of the distributional nature of word vectors, proximities on the two-dimensional map make no distinction between synonyms and antonyms.



(a) positive qualities

(b) negative qualities



(c) attitudes and mindsets

(d) physiological stimuli

Figure 3: Close-ups from Figure A.1

The interpretation of Figure A.1 is perhaps too daunting because of the accumulation of adjectives scattered across the Euclidean space. To facilitate interpretation, I partitioned the data points into classes based on word vectors with k-means clustering. Figure A.2 (Section A)

displays 10 partitions, each of which has its own color. On the whole, the *k*-means partitions are consonant with the t-SNE clusters as evidenced by the presence of neat color groupings. There are some exceptions, however. For example, *junior* (top of Figure A.1, in red) belongs to the same *k*-means partition as *young(er)* (upper right corner of the plot) because it denotes an age-related quality. Its position on the two-dimensional map reveals one dimension that *k*-means clustering ignores: its distributional proximity with other adjectives such as *leading* or *major*.

To see how manual annotation correlates with both t-SNE and *k*-means clusterings, I over-plotted the semantic classes obtained from manual annotation, as shown in Figure A.3 (Section A). The semantic classes are scattered across the plot. For example, adjectives that denote a dimension or a position in time or in space are found in the left, upper left, lower, and rightmost parts of the plot. Because there are 59 semantic classes and only 10 *k*-means classes, we have no a priori reason to expect them to match. To explore the association between semantic and *k*-means classes, I ran a correspondence analysis, whose graphic output is displayed in Figure 4. Interestingly, consistent clusters emerge. For example, negative qualities such as `dullness`, `discomfort`, `psych_stim_bad`, `physical_property_bad`, `value_undesirable`, etc. tend to cluster in the lower left corner of the plot along with *k*-means classes `1` and `4`. Conversely, positive qualities such as `energy_good`, `psych_stim_good`, `soc_psych_prop_good`, `physical_property_good`, or `value_desirable` cluster in the upper left corner along with classes `6`, `9`, and `10`. Neutral or ordinary properties tend to cluster in the upper right corner along with classes `5` and `8`. Finally, modal properties are likely to be found in the lower right corner along with classes `2`, `3`, and `7`. This means that vector-based *k*-means classes can be used as a proxy for semantic classes, providing that one is willing to forgo conceptual precision to the benefit of categories based on distributional coincidence.

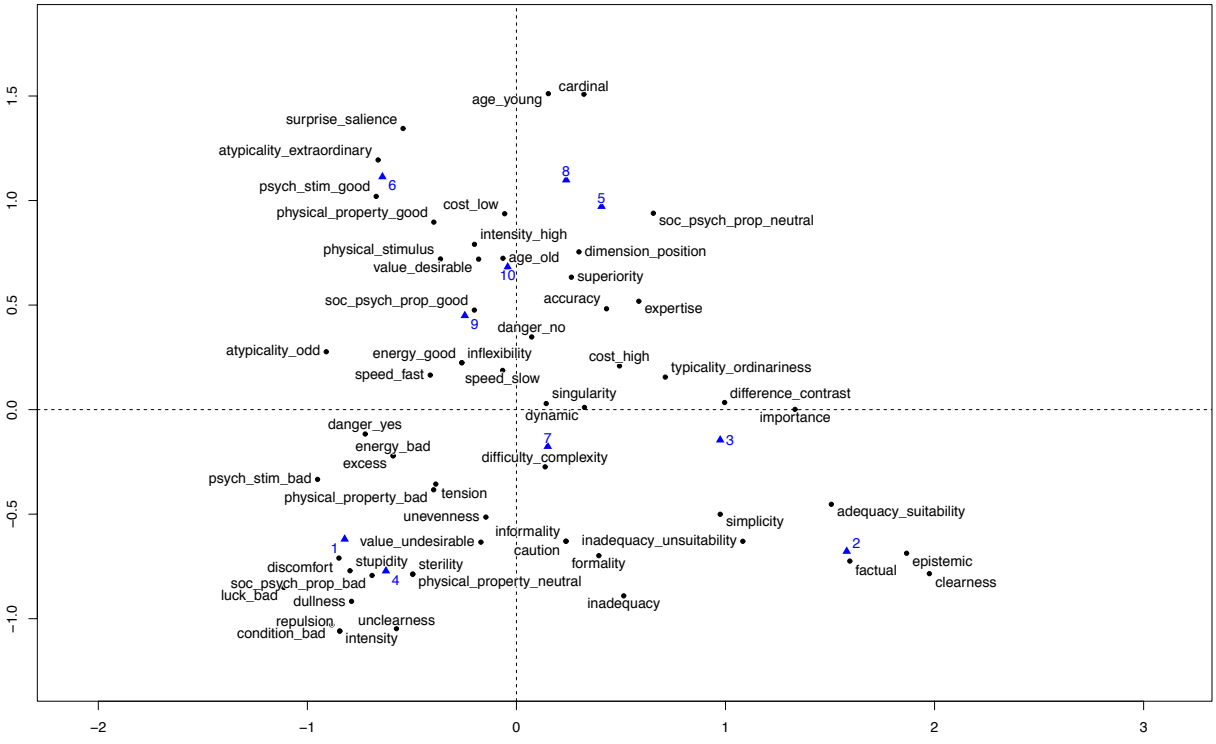


Figure 4: A correspondence analysis plot showing the distribution of semantic classes (black) with respect to *k*-means classes (blue)

All in all, word-vector clusters and 'manual' semantic classes do not overlap neatly. This

is hardly surprising because the assignment of semantic classes is 'supervised' by definition, whereas vector-based machine learning is radically unsupervised. However, the above shows that they are globally compatible.

5 Discussion

State-of-the-art word-vector semantics is definitely promising. There are, however, several areas in which it has yet to improve. I briefly discuss two below.

5.1 The one-vector-per-word problem

One structural problem that corpus linguists need to address before benefiting fully from word vectors in semantic annotation task is the way that polysemy and homonymy are currently handled. WordNet 3.1 lists 21 possible meanings for *hot* such as "physical heat", "characterized by violent and forceful activity or movement", "bold and intense" (colors), "sexually excited", "recently stolen", "very popular or successful", etc. It is indeed a very polysemous adjective. Typically, a vector matrix assigns one vector representation per word. All the meanings derived from the word distribution in the corpus are conflated into a single string of real numbers. In other words, if we are to annotate a dataset of adjectives, we are going to have to accept that the vector representation of *hot* is unique. Given that the vector keeps track of all the different contexts in which the adjective appears, and given that the senses of *hot* are somewhat related, corpus linguists may arguably accept that a polysemous word is given a unique word-vector representation.

A bigger issue arises with homonyms, i.e. words with identical form but several unrelated meanings such as *bank* (a financial institution vs. the slope beside a body of water). In this case, conflating unrelated meanings into a single vector makes little sense. It is hard to avoid this issue without making sure that the target word receives a specific annotation depending on the contexts where it appears, e.g. *bank_1* for the financial institution and *bank_2* for the slope, as shown in (5) and (6).

(5) The central **bank_1** tried to stem the panic when it increased interest rates from 10.5 percent to 17 percent on Dec. 16. (*The Washington Post*)

(6) He stood on the muddy **bank_2** at the side of the Mississippi and shrugged, looking at the faces around him. (Heather Graham, *Deadly Night*)

This preliminary supervised step requires that the different contexts be distinguished neatly from the start in the training corpus. However, it undermines one of the strongest selling points of word vectors: unsupervised learning.

5.2 Multiword expressions

MWEs are strings of two or more lexemes that are idiosyncratic in some respect. They are challenging at both the detection and comprehension levels. The grammatical status of MWEs has been an issue at least since the "rules vs. the lexicon" debate (Rumelhart & McClelland 1986; Langacker 1987; Pinker & Prince 1988; Pinker 1999). Because rules capture all the regularities in language, MWEs should have no place in the grammar proper because they are lexical. Because the lexicon consists of words or morphemes, it does not include MWEs because they are phrasal. Jackendoff (1997, chapter 7) advocates the inclusion of "phrasal lexical items"

(i.e. "lexical items larger than X^0") in the lexicon. An alternative, although related, solution proposed by construction grammar approaches delegates MWEs to a "constructicon" (Goldberg 2006, 64). Grammar consists of a large inventory of constructions, varying in size and complexity, and ranging from morphemes to fully abstract phrasal patterns (Goldberg 2003).

Such complex strings are frequent. Sag, Baldwin, Bond, Copestake, and Flickinger (2002) estimate that 41% of the entries in WordNet 1.7 are MWEs. MWEs assume a wide range of forms such as:

- institutionalized phrases and clichés (*love conquers all*, *no money down*);
- idioms (*kick the bucket*, *sweep under the rug*);
- fixed phrases (*by and large*);
- compound nouns (*black and white film*, *frequent-flyer program*);
- verb-particle constructions (*eat/look/write up*);
- light-verb constructions (*have a drink/*an eat*, *make/*do a mistake*);
- named entities (*San Francisco*);
- lexical collocations (*telephone box/booth/*cabin*, *emotional baggage/*luggage*).

MWEs are easily mastered by native speakers. Yet, their linguistic status is still problematic and their detection and interpretation still pose a major challenge for NLP techniques.

Suppose you investigate *quite* and *rather* constructions. The typical solution is to treat the MWE as "words-with-spaces" and concatenate the words in the syntactic pattern in which they are found: e.g. *quite*_a/*rather*_a vs. a_*quite*/a_*rather*. Then, one determines the vector profile of the whole phrase. Although this might work for intensifiers, as well as institutionalized phrases, clichés, and fixed phrases, it will fail to detect the other types of MWEs satisfactorily for a wide variety of reasons, some being grammatical, others being referential (Sag et al. 2002).

Grammatical variation can be exemplified by the inflectional variation of idioms (*kicking/kicked/will kick the bucket*) or compound nouns (*editor(s)-at-large*), the reflexivity of some verbs like *wet* (*wet myself/yourself/himself/herself/themselves*/etc.), and the long-distance dependencies at work in verb-particle constructions (*look up the Yeti* is ambiguous between "glance up at the Yeti" and "check the dictionary definition of the word *Yeti*"). Referential variation is witnessed at the level of named entities. A baseball team such as *the Oakland A's* can be referred to as *Oakland* (which must be distinguished from the city) or *the A's* (which must be distinguished from the letter of the alphabet).

Light-verb constructions pose a different kind of problem. They are characterized by erratic selectional preferences. For example, *take* is a light verb in the following contexts:

(7) You should take a shower.

(8) I need to take a nap/lie-down.

(9) Take a break.

(10) He has taken advantage of the situation.

(11) Take a look.

(12) Take a chance.

Take is a plain lexical verb in other contexts:

(13) Please, take a cookie/candy.

(14) I'll take the first plane.

(15) I am going to take a coach.

In her study of the <*have* + *a* + deverbal noun> construction, Wierzbicka (1982) argues against its idiomaticity and for the existence of "strict semantic rules". Specifically, Wierzbicka finds that the action profiled by the construction is "agentive, experiencer-oriented, antidurative, atelic, and reiterative" (p. 759). Because *bite* fulfills all these conditions and *eat* does not, the construction will select the former and not the latter. However, usage tells us that *have an eat* is attested, in certain contexts, especially when the social function of *eat* is foregrounded as in (16), or when the durativity of the process is cancelled by an adjective as in (17).

(16) Let's have an eat out!

(17) I would go home and have a quick eat (...). (T.S. Eliot, *Silas Marner*)

Because usage is flexible, a dictionary of deverbal nouns cannot be used to detect light verbs accurately. In this specific case, the lexical proliferation problem dramatically skews the ratio towards recall to the detriment of precision.

Finally, MWEs are hard to detect because they often contain closed-class words. Such words are very frequent. To accommodate phrases in a vector-space model, Mikolov, Sutskever, Chen, Corrado, and Dean (2013b) propose a detection technique that consists in subsampling closed-class words such as determiners and prepositions, which easily occur millions of time in any large corpus. Such words are generally considered meaningless with respect to the less frequent open-class words. The problem with this subsampling technique is that it may discard the closed-class words that are often part of idiomatic constructions (e.g. *at* in *congressman/editor at large*, or *the* in *kick the bucket*). The inventory of detection issues regarding MWEs is vast, and certainly not exhausted here.

Once solved, other problems emerge in the field of vector-based MWE comprehension. Non-compositonality is foremost among them (Padó & Lapata 2007). Given *red*, which denotes a color, and *tape*, which denotes a long thin piece of cloth, paper, or plastic, it is unlikely that a machine will relate *red tape* to a needlessly time-consuming bureaucratic procedure, unless the machine has been instructed by the researcher to treat the two words as a single chunk. This is because *red tape* is non-compositional. Let W_1 and W_2 be the two lexical constituents (*red* and *tape*) of the nominal compound N (*red tape*). The syntax-dependent composition function yielding a nominal compound, adapted from Mitchell and Lapata (2010) and Dinu and Baroni (2014), should be:

$$\vec{N} = f_{comp}(\vec{w}_1, \vec{w}_2), \tag{i}$$

where $\vec{w}_1$ and $\vec{w}_2$ are the vector representations associated with W_1 and W_2. Dinu and Baroni (2014) and Mikolov et al. (2013b) have found that composition can be defined as the application of linear transformations to the two constituents by summing up their respective vectors:

$$f_{comp}(\vec{w}_1, \vec{w}_2) = \vec{w}_1 + \vec{w}_2. \tag{ii}$$

This equation is but a baby step in the resolution of (non-)compositionality issues. Right now, compositionality is an open area of active research. It is a recurring topic of research for SIGLEX, the Special Interest Group on the Lexicon of the Association for Computational Linguistics. SIGLEX is the umbrella organization for SemEval (Semantic Evaluation), which is an ongoing series of evaluations of computational semantic analysis systems. Among the major

tasks in semantic evaluation, compositionality is part of word sense disambiguation. Interestingly, human performance is the gold standard to evaluate the performance of NLP systems in semantic evaluation. This suggests that, right now, exclusively unsupervised algorithms, however powerful, have some way to go before they outperform manual annotation.

6 Conclusion

As pointed out by Gries (2008), "corpus linguistics is all about distributional data". In this regard, word vectors are a timely addition to the existing apparatus of techniques aimed at capturing similarities between words. Word vectors have important implications for solving semantic annotation issues. They suggest decisive opportunities for future research in corpus building, dataset compilation and, more generally, corpus-based semantics.

However, although it has made and is still making considerable progress, word-vectors semantics based on deep learning is still in its infancy. Chances are that, by the time this article is published, some limitations I have mentioned will have been addressed already.

References

Baayen, R. H. (2001). *Word Frequency Distributions*. Dordrecht: Kluwer Academic Publishers.

Baroni, M., Bernardini, S., Ferraresi, A., & Zanchetta, E. (2009). The WaCky Wide Web: A collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, *43*(3), 209–226.

Baroni, M., Dinu, G., & Kruszewski, G. (2014). Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *ACL (1)* (pp. 238–247).

Boas, H. C. & Schuchard, S. (2012). A corpus-based analysis of preterite usage in Texas German. In *Proceedings of the 34th Annual Meeting of the Berkeley Linguistics Society*.

Burnard, L. (2000). Reference Guide for the British National Corpus (World Edition). Web Page. Retrieved from http://www.natcorp.ox.ac.uk/archive/worldURG/urg.pdf

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, *20*, 37–46. doi:10.1177/001316446002000104

Cohen, J. (1968). Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*, *70*(4), 213–220. doi:10.1037/h0026256

Davies, M. (2013). Corpus of global web-based english: 1.9 billion words from speakers in 20 countries. Retrieved from http://corpus.byu.edu/glowbe/

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, *41*(6), 391.

Desagulier, G. (2015). Forms and meanings of intensification: A multifactorial comparison of *quite* and *rather*. *Anglophonia*, *20*. doi:10.4000/anglophonia.558

Dinu, G. & Baroni, M. (2014). How to make words with vectors: Phrase generation in distributional semantics. In *ACL (1)* (pp. 624–633).

Divjak, D. & Fieller, N. (2014). Cluster analysis: Finding structure in linguistic data. In D. Glynn & J. Robinson (Eds.), *Corpus Methods for Semantics: Quantitative Studies in Polysemy and Synonymy* (pp. 405–441). Amsterdam: John Benjamins.

Dixon, R. M. W. & Aikhenvald, A. Y. (2004). *Adjective Classes: A Cross-Linguistic Typology*. Oxford: Oxford University Press.

Firth, J. R. (1957). A synopsis of linguistic theory 1930-55. In *Studies in Linguistic Analysis (special volume of the Philological Society)* (Vol. 1952-59, pp. 1–32). Oxford: The Philological Society.

Glynn, D. (2010). Testing the hypothesis: Objectivity and verification in usage-based cognitive semantics. In D. Glynn & K. Fischer (Eds.), *Corpus-Driven Cognitive Semantics. Quantitative Approaches* (pp. 239–270). Berlin: Mouton de Gruyter.

Goldberg, A. E. (2003). Constructions: A New Theoretical Approach to Language. *Trends in Cognitive Sciences*, *7*(5), 219–224.

Goldberg, A. E. (2006). *Constructions at Work: The Nature of Generalization in Language*. Oxford & New York: Oxford University Press.

Gries, S. T. (2008). Dispersions and adjusted frequencies in corpora. *International Journal of Corpus Linguistics*, *13*(4), 403–437.

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

Hollmann, W. B. & Siewierska, A. (2007). A construction grammar account of possessive constructions in Lancashire dialect : some advantages and challenges. *English Language and Linguistics*, *11*(2), 407–424. doi:10.1017/S1360674307002304

Jackendoff, R. (1997). *The Architecture of the Language Faculty*. Cambridge, Mass. ; London: MIT Press.

Langacker, R. W. (1987). *Foundations of cognitive grammar*. Stanford: Stanford University Press.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). Efficient Estimation of Word Representations in Vector Space. *CoRR*, *abs/1301.3781*. Retrieved from http://arxiv.org/abs/1301.3781

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems* (pp. 3111–3119).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013c). Distributed Representations of Words and Phrases and their Compositionality. *CoRR*, *abs/1310.4546*. Retrieved from http://arxiv.org/abs/1310.4546

Mikolov, T., Yih, W.-t., & Zweig, G. (2013d). Linguistic regularities in continuous space word representations. In *Proceedings of NAACL-HLT* (pp. 746–751). Retrieved from http://www.aclweb.org/anthology/N/N13/N13-1090.pdf

Miller, G. A. & Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes*, *6*(1), 1–28.

Mitchell, J. & Lapata, M. (2010). Composition in distributional models of semantics. *Cognitive science*, *34*(8), 1388–1429.

Padó, S. & Lapata, M. (2007). Dependency-based construction of semantic space models. *Computational Linguistics*, *33*(2), 161–199.

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)* (pp. 1532–1543). Retrieved from http://www.aclweb.org/anthology/D14-1162

Piao, S., Bianchi, F., Dayrell, C., D'Egidio, A., & Rayson, P. (2015). Development of the multilingual semantic annotation system. In *The 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies* (pp. 1268–1274). Association for Computational Linguistics.

Pinker, S. (1999). *Words and Rules: The Ingredients of Language*. New York: Basic Books.

Pinker, S. & Prince, A. (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition*, *28*(1-2), 73–193.

Rong, X. (2014). Word2vec parameter learning explained. *CoRR*, *abs/1411.2738*. Retrieved from http://arxiv.org/abs/1411.2738

Rumelhart, D. E. & McClelland, J. L. (1986). Parallel distributed processing: Explorations in the microstructure of cognition, vol. 2. In D. E. Rumelhart, J. L. McClelland, & C. PDP Research Group (Eds.), (Chap. On Learning the Past Tenses of English Verbs, pp. 216–271). Cambridge, MA, USA: MIT Press. Retrieved from http://dl.acm.org/citation.cfm?id=21935.42475

Sag, I. A., Baldwin, T., Bond, F., Copestake, A., & Flickinger, D. (2002). Multiword expressions: A pain in the neck for nlp. In *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 1–15). Springer.

Schütze, H. (1992). Dimensions of meaning. In *Supercomputing'92., proceedings* (pp. 787–796). IEEE.

van der Maaten, L. & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*, 2579–2605.

Wierzbicka, A. (1982). Why can you have a drink when you can't **have an eat*? *Language*, 753–799.

Zeldes, A. (2012). *Productivity in Argument Selection: From Morphology to Syntax*. Berlin & New York: Mouton de Gruyter.

Zipf, G. K. (1949). *Human Behavior and the Principle of Least Effort*. Cambridge: Addison-Wesley.

A Appendix

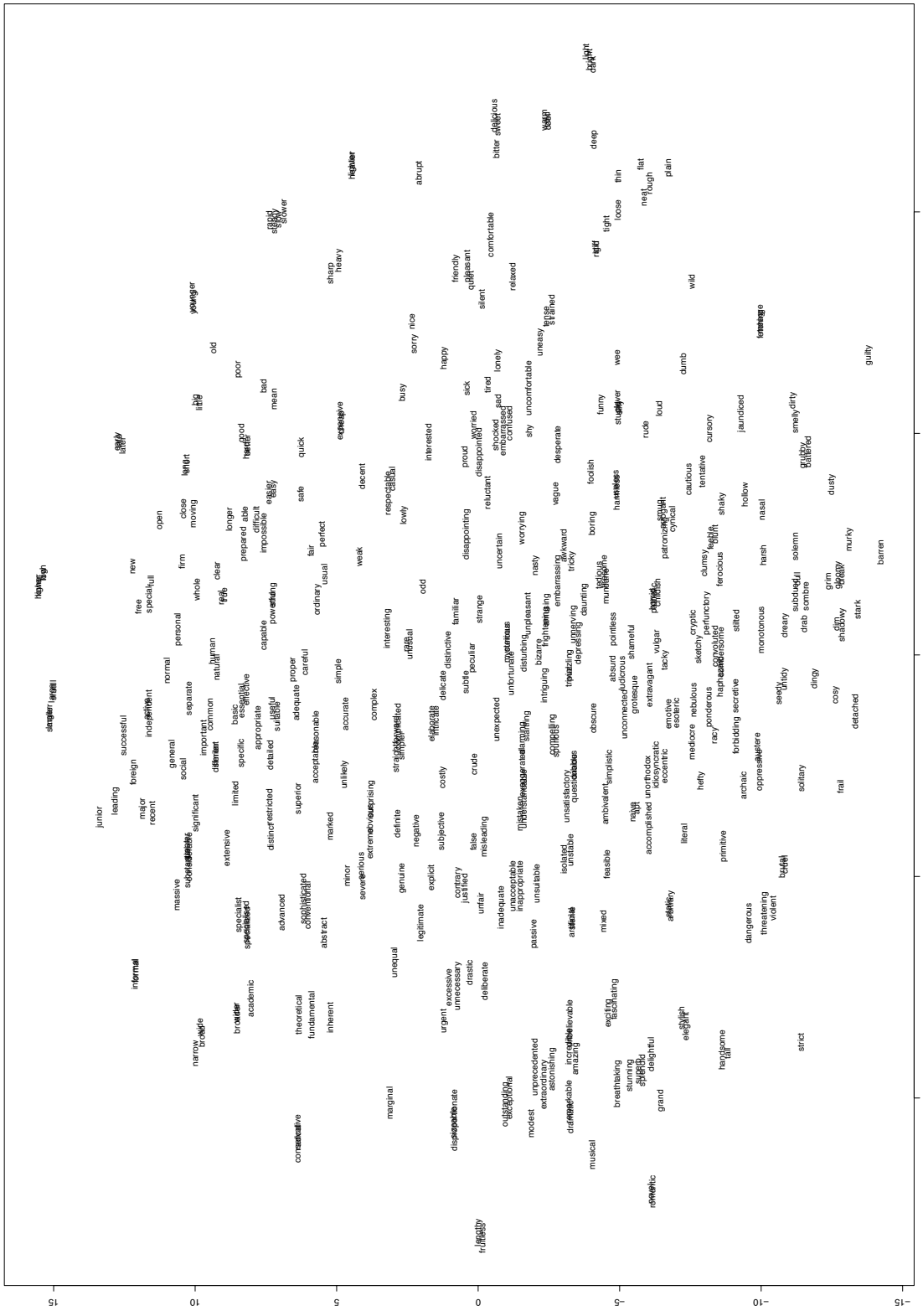


Figure A.1: A cluster plot of adjectives based on 50-dimensional word vectors with Barnes–Hut–SNE

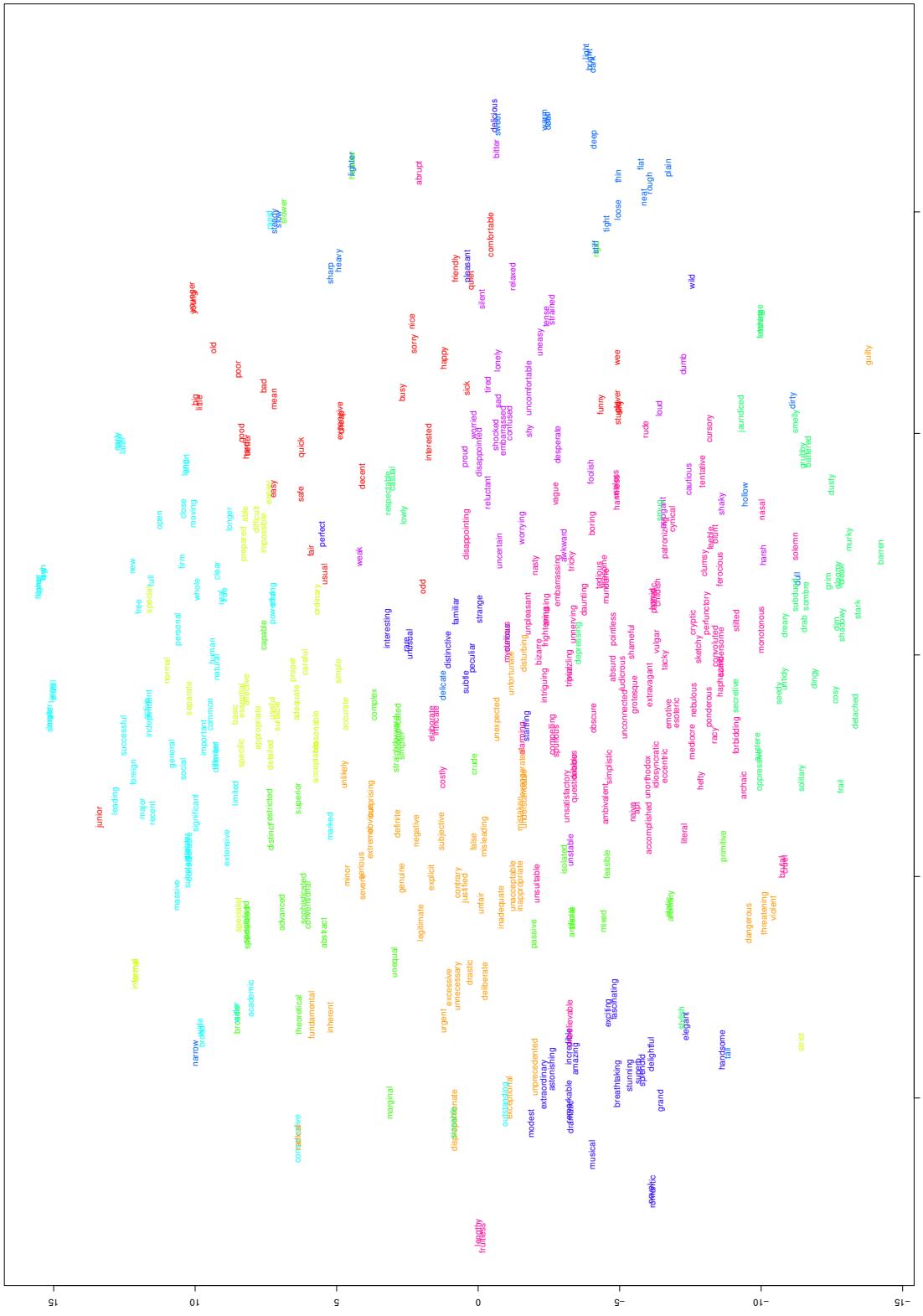


Figure A.2: A cluster plot of adjectives based on 50-dimensional word vectors with Barnes–Hut–SNE (with k-means clustering)

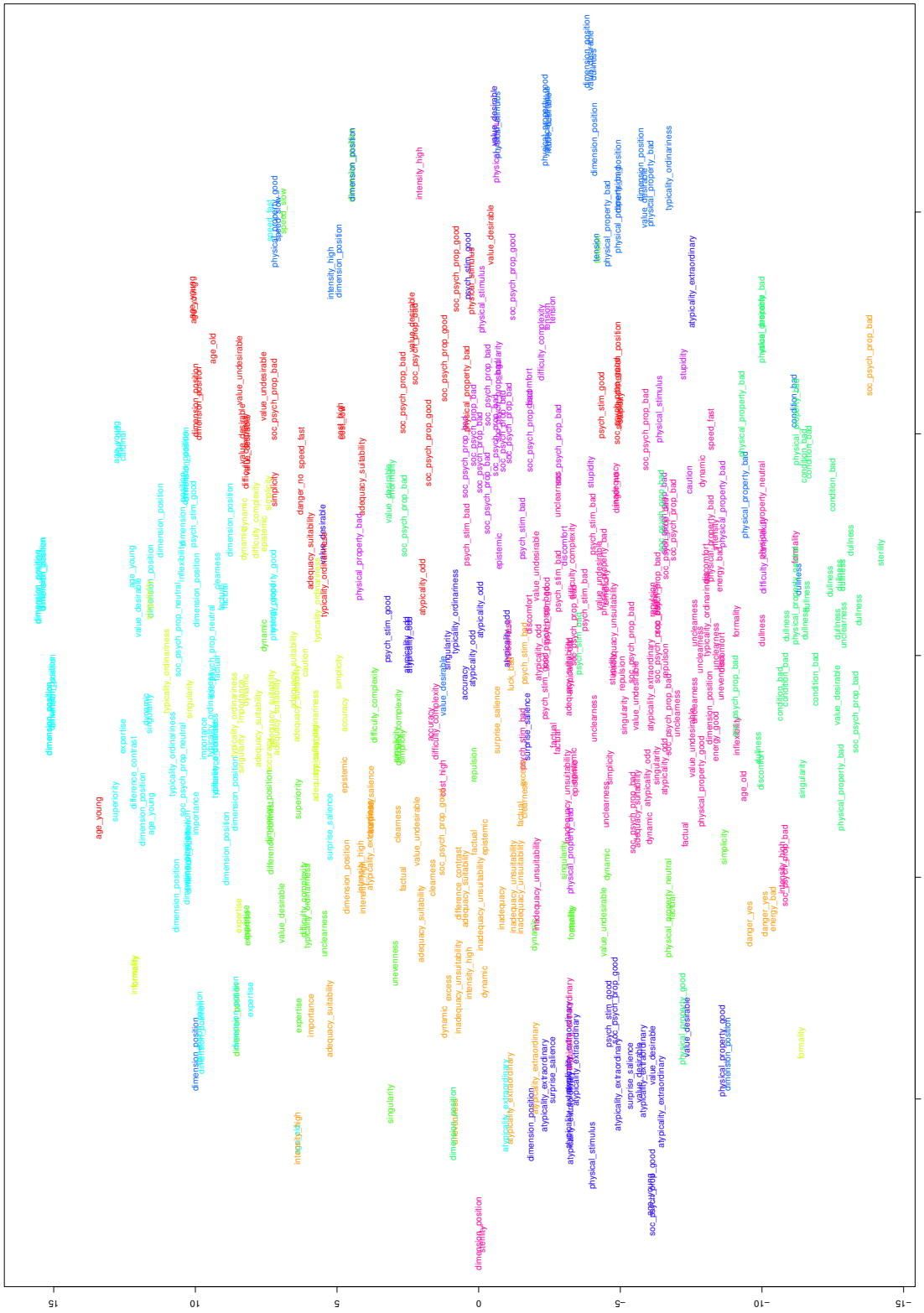


Figure A.3: A cluster plot of semantic classes based on 50-dimensional word vectors with Barnes–Hut–SNE