Random vector generation of a semantic space

Jean-François Delpech    Sabine Ploux
Institut des Sciences Cognitives
UMR5304 CNRS - Université de Lyon
67, boulevard Pinel
69675 BRON cedex, France
jfdelpech@gmail.com    sploux@isc.cnrs.fr

We show how random vectors and random projection can be implemented in the usual vector space model to construct a Euclidean semantic space from a French synonym dictionary. We evaluate theoretically the resulting noise and show the experimental distribution of the similarities of terms in a neighborhood according to the choice of parameters. We also show that the Schmidt orthogonalization process is applicable and can be used to separate homonyms with distinct semantic meanings. Neighboring terms are easily arranged into semantically significant clusters which are well suited to the generation of realistic lists of synonyms and to such applications as word selection for automatic text generation. This process, applicable to any language, can easily be extended to collocations, is extremely fast and can be updated in real time, whenever new synonyms are proposed.

1. Introduction

In their seminal work, Ploux and Victorri\textsuperscript{1} have used synonymy relations deduced from French electronic dictionaries to create semantic spaces around French words and their neighbors. Their definition of “synonymy” is fairly broad and includes hyponymy (moineau and oiseau), hyperonymy (arme and pistolet) or even non-synonymous, but related terms (autocar and automobile); however, in their work, true synonyms (i.e. terms which are more or less interchangeable) form cliques of the graph of synonyms, i.e. maximally complete subgraphs. While this is very interesting from a theoretical standpoint, as it then becomes straightforward to evaluate an interclique distance (or degree of separation) between any two terms in the graph (as long as neither belongs to an island, such as lapereau and lapinot), it is not very useful in practice. For example, an author in search of the right term may well not be interested in strict synonyms; terms with related or even opposed meanings can often be preferable in rhetorical figures. Also, in many applications such as automatic text generation, a well-defined and mathematically well behaved semantic distance between terms is often a prerequisite.

In this report, we show how an Euclidean semantic distance can quickly and easily be constructed from Ploux and Victorri’s database (which contains 54,685 terms and 116,694 cliques).

2. Construction of a semantic space

2.1. The vector space model

Since the pioneering work of Salton\textsuperscript{2,3}, it is well understood that any combination of terms, such as a clique, can be seen as a vector in a space where each dimension represents a distinct term (or lemma.)

\begin{equation}
C_j = (t_{1,j}, t_{2,j}, \ldots, t_{d,j})
\end{equation}

This representation is extremely fruitful and forms the basis of numerous information retrieval systems; it suffers however from a severe limitation in that each term is orthogonal to each other. Of course, the dual equation from Equation 1,

\begin{equation}
T_k = (c_{1,k}, c_{2,k}, \ldots, c_{d,k})
\end{equation}
may be used to compute term distances (or similarities) but the very high dimensionality of the subtending space makes such distances difficult to compute and to interpret: this is the “curse of dimensionality”.

2.2. Overlap similarity between terms

If \( D_i \) with cardinality \( d_i \) is the set of distinct terms occurring in all the cliques containing term \( t_i \), we define the overlap similarity between two terms as the cardinality \( d_{i,k} \) of the intersection \( D_i \cap D_k \) (each word being counted only once.) Obviously, \( d_{i,k} = 0 \) for most \((i,k)\) pairs, since for any \( i \) the total number of distinct terms in the database is much larger than \( d_i \), which ranges from 2 to 243 in Ploux and Victorri’s database with an average value of 8.5.

2.3. Contextonyms

“Contextonyms” are words which co-occur in a given context (as for example in the same sentence of a corpus); while they are not synonyms, they are obviously closely related. Ji, Ploux and Wehrli have proposed an automatic contextonym organizing model (ACOM) which relies on counting of co-occurrences and evaluating their probabilities to automatically produce and organize contextonyms for a target word. The test results, after training on an English corpus maintained by Project Gutenberg, show that the model is able to classify contextonyms as well as to reflect words' minute usage and nuance.

2.4. Latent semantic indexing

Dimensionality reduction can be achieved by a low-rank approximation of the term-document matrix. This can be done by Latent Semantic Indexing (LSI), which reduces dimensionality through a singular value decomposition (SVD) of the term-document matrix, retaining only a comparatively small number of the largest singular values. This method has been very successfully used for document indexing and retrieval. It suffers nevertheless from limitations:

- SVD is computationally intensive, even though the large term-document matrix is very sparse;
- There is no really satisfactory way to increment the results as new terms/documents become available.

More importantly, it is not well suited to generating a semantic space from cliques. The resulting, lower dimensional space is the best approximation, in the least squares sense, of the position of any term belonging to the whole set of cliques: the distance between any pair of terms will be optimal, while what is really of interest from the present perspective is the accurate determination of distances between semantic neighbors.

As a test, a SVD decomposition of the clique-term matrix (of which eq. 2 is a row) was performed. It reduced the matrix size from 54,685 x 116,694 to 54,685 x 250, meaning that each term was associated with a vector having 250 orthogonal components. The decomposition, which took 134 sec. on a desktop computer, was clearly unsatisfactory as the singular values decayed very slowly, from 63.31 for the first coordinate to 37.4 for the 250th one. According to this computation, the first few neighbors of rapsoode would be rimailleur, rimeur, versificateur, méromane, fils d’Apollon, favori des Muses, favori du Parnasse, nourisson des Muses, héros du Pinde, mâché-laurier, all with similarities extremely close to 1.0. While clearly in the right neighborhood, this seems to be of limited usefulness; note however that in practice a restriction to the first two or three largest singular values may often yield useful information.

2.5. Neural networks

Word order is not considered in this publication, but it should be mentioned for completeness that neural networks are often used in natural language processing to encode word sequences (see for an extended review and bibliography). In a recent publication, Mikolov et al. have introduced two novel model architectures for computing continuous vector representations of words from very large data sets. They report large improvements in accuracy at a computational cost which is still substantial, but that they claim is much lower than previous architectures. An interesting consideration is that, according to Mikolov et al., the learned vectors explicitly encode many linguistic regularities and patterns.
2.6. Random vectors and random projection

2.6.1. Remarks on high-dimensionality spaces

Dasgupta\cite{Dasgupta1999} points out that intuitions valid in a low-dimensionality space may be totally misleading in a high-dimensionality space. For example, a set of points picked at random from the unit ball

$$\{x \in V : \|x\| < 1\}$$

(3)

will have some significant fraction near the origin, say within distance $1/2$ if $d = 3$, but this fraction becomes rapidly vanishingly small as the dimension $d$ becomes large; for example for $d = 250$, $d^{250} = 0.5^{250} \approx 5.5 \times 10^{-76}$.

Another useful remark is that while obviously one cannot create more than $d$ orthogonal vectors in a space of dimension $d$, one can create an exponentially large number of vectors quasi-orthogonal to each other; in other words\cite{Dasgupta1999}, a set of $\exp(O(e^2d))$ vectors picked at random will with high probability be quasi-orthogonal, i.e. have angles of $90 \pm \epsilon$ with each others. The seed vectors referred to below will be selected from such a set $S_d$.

While an orthogonal projection will in general reduce the average distance between points, it is also known, as shown by Johnson and Lindenstrauss in an often cited paper\cite{Johnson1984}, that distances may be almost perfectly preserved for any $n$ points in an arbitrary number $d$ of dimensions when projected to a random subspace of $O(\log n)$ dimension.

2.6.2. Building random vectors

The comparatively recent method of random projection\cite{Indyk2001,Charikar2002,Larson1996} is based on these three preceding remarks and proceeds as follows:

1. Uniquely associate with each term $t_i$ a random seed vector $s_i \in S_d$ having $d$ independent coordinates;
2. Associate with each clique $c_k$ the vector $C_k = \sum_{i \in c_k} \rho_i^k s_i$ where $i \in c_k$ refers to the set of terms found in clique $c_k$ and $\rho_i^k$ is a function of the number of occurrences in $c_k$ of term $t_i$ and of the weights associated with $t_i$;
3. Finally, associate with each term $t_i$ the (suitably weighted) sum of the vectors of the cliques in which $t_i$ appears, $T_i = \sum_{k \ni t_i} C_k$ where $k \ni t_i$ refers to the cliques containing $t_i$. In what follows, we shall assume without loss of generality that the term vectors $T_i$ are normalized to unity.

Obviously, each term vector $T_i$ is now embedded in a $d$-dimensional Euclidean semantic space and the similarity $\sigma_{ij}$ between terms $t_i$ and $t_j$ is the scalar product of the associated term vectors:

$$\sigma_{ij} = \langle T_i, T_j \rangle$$

(4)

It is easy to see that $\sigma_{ij}$ ranges from -1 to 1. It is sometimes more convenient to consider the distance $\mathcal{D}_{ij}$ which is related to the similarity by $\mathcal{D}_{ij} = \sqrt{2(1 - \sigma_{ij})}$ and ranges from 0 (same $t_i$ and $t_k$) to 2 (exactly opposite terms; note however that owing to the extreme sparsity of a high-dimensional space, the neighborhood exactly opposite a term is in practice always empty.)

2.6.3. Locality property

Building a term vector $T_i$ by the process described above involves only the terms pertaining to the set $D_i$, defined in section 2.2. It is thus a purely local process: updating the semantic space requires only a few ten or a few hundred operations, orders of magnitude less than its initial generation (provided small changes to the weights are neglected, which is usually acceptable as they are logarithmic in term frequency and inverse document frequency.)

This does not imply that the similarity of term $t_i$ with term $t_k \not\in D_i$ is zero, even though $t_k \not\in D_i \implies t_i \not\in D_k$, since $t_i$ and $t_k$ may well have neighbors in common. For example carotte and fraude have a degree of separation of 2 but a similarity of 0.364.

The seed vectors $s_i$ are not quite orthogonal and the scalar product $\langle s_i, s_j \rangle$ will usually be small, but not zero. Thus, even for uncorrelated term vectors $T_i$ and $T_j$, their similarity $\sigma_{ij}$ will usually be small but non-zero. This induce an unavoidable noise which is studied below in some detail.

3/10
2.7. Practical implementation

A normalized seed vector embedded in a $d$-dimensional space has $d$ coordinates of which $d - 2m$ are 0, $m$ are $+1/\sqrt{2m}$ and $m$ are $-1/\sqrt{2m}$. Having the same number of positive and negative coordinates ensures that the scalar product of two seed vectors is 0 on the average. As seed vectors need to be very close to orthogonal with each others, the number $2m$ of non-zero coefficients must be substantially smaller than the dimension $d$.

The number of available, distinct seed vectors is the product of the number of combinations $\binom{d}{2m}$ of $2m$ non-zero coordinates amongst $d$ coordinates, times the number of ways $\binom{2m}{m}$ of distributing $m$ positive and $m$ negative coordinates amongst these non-zero coordinates:

$$N_{seed}(d, m) = \binom{d}{2m} \times \binom{2m}{m}$$  \hspace{1cm} (5)

In practice, $N_{seed}$ should be much larger than the number of distinct terms to guarantee a negligible collision probability (i.e. two distinct terms having the same seed vector). This condition is already amply met with $m \geq 4$ as $N_{seed}(250, 4) \approx 2.4 \times 10^{16}$ when a dimension $d = 250$ is selected.

Given $d$ and $m$ and noting for simplicity $p = 2m/d$, the probability $P_{overlap}(v, d)$ of an overlap $v$ between two randomly selected seed vectors is

$$P_{overlap}(v, d) = \binom{2m}{v} \times p^v \times (1 - p)^{2m-v}$$  \hspace{1cm} (6)

When two randomly selected, normalized seed vectors have an overlap of $v$ non-zero coordinates, their scalar products will be arranged symmetrically around zero and vary in discrete steps of $1/2m$. An overlap of 1 will generate the two scalars $1/2m$ and $-1/2m$ with probabilities 1/2, an overlap of 2 will generate $2/2m$ with probability 1/4, 0 with probability 1/2, and $-2/2m$ with probability 1/4; more generally, an overlap $v$ will generate the scalar $s/2m$ with the probability

$$P_{scalar}(v, s) = \binom{v}{q_s} \times 2^{-v}$$  \hspace{1cm} (7)

where the factors $q_s = (v + s)/2$ are restricted to integer values and by virtue of the identity $\sum_{k=-v}^{v} \binom{k}{q_s} = 2^v$.

It can be seen from equations (6) and (7) that the noise decreases more or less linearly with $d$. Theoretical and experimental scalar products of two seed vectors, as computed from equations (6) and (7), are plotted in the next figure (next page) where:

- The light vertical lines are increments of $0.01\sigma$, the heavier vertical lines are at 0, $-0.1\sigma$ and $+0.1\sigma$.
- The two horizontal lines are at 1.0 and 0.136.
- The red dots are computed by taking the scalar products of 1,000,000 'term vectors' each synthesized by the addition of 5 random seed vectors. If instead we do the statistics directly on seed vectors, the result is unchanged except that the dots now occur only at multiples of 0.01 and the total is accordingly 5 times larger.
- The black dots are statistics over 40,000 points, starting with 10,000, taken from the tail of the neighbors of an arbitrary term (here rapsode.)
- The purple dots are a Gaussian with a standard deviation of 0.063.
3. Results

Obviously, the size of the database of the term vectors $\mathcal{T}_i$ is itself linearly dependent on the dimension; for $d = 2500$, each vector occupies 10 kB if a single coordinate is represented by a 4-byte floating point number. A database of 1,000,000 distinct terms would thus occupy 10 GB with this elementary data structure; however, for many applications, $d = 250$ will be sufficient and/or more sophisticated data structures may be implemented. Computation times will also increase more or less linearly with $d$, because they mostly involve stepping through all the dimensions.

3.1. Term neighbors

The following four figures have been constructed by compiling eight independent databases of 54,685 term vectors for each of the four indicated $(m, d)$ couples; tf-idf statistical weights were used. Typically, on a small desktop computer, the compilation time is 3 to 4 seconds for $d = 250$ and 20 to 30 seconds for $d = 2500$.

Abscissas are proportional to the logarithm of the neighbor's rank (From 1 for maison in the upper right corner to 1000 in the lower left corner) and ordinates are the similarities to maison. For a given neighbor, the horizontally aligned red dots represent the eight scalars $\sigma_j$ computed from the eight databases and the thicker, black dot is the average value $\sigma_{avg}$ of the eight scalars. The neighbors are arranged in non-increasing order of their $\sigma_{avg}$ with maison.

Even though the diameter $d_{maison}$ of maison as defined in section 2.2 is only 98, there are several hundred significant neighbors: while most close neighbors belong to the set $\mathcal{D}_{maison}$, many lie at more than one degree of separation from each others (their cliques are separated by more than one vertex). Also, it can be seen, as expected, that the noise is inversely proportional to $\sqrt{d}$ but not very dependent on $m$ (in fact, the only noticeable effect of a lower $m$ is that there are more outliers) and that a standard deviation of about 0.063 is not unrealistic for $d = 2500$. 

5/10
In what follows, unless otherwise noted, we'll use \( d = 2500 \) and \( m = 50 \). The size of the file containing the 54,685 term vectors is then 547,724,964 bytes, including some overhead. The number of available seed vectors being \( \approx 2.0 \times 10^{28} \), the risk of collision is totally negligible.

The 100 first neighbors of *maison* are listed by decreasing similarity in table 1 next page. It is clear that the proximity decreases with \( \sigma \), but that those neighboring words are all reasonably close to *maison* in its various meanings.

| From 1 to 20 | From 21 to 40 | From 41 to 60 | From 61 to 80 | From 81 to 100 |
|--------------|--------------|--------------|--------------|--------------|
| 1 maison     | 21 chez-soi  | 41 ménage    | 61 ermitage  | 81 domestique|
| 2 demeure    | 22 cassine   | 42 bâtiment  | 62 appartement| 82 plaque_de_blindage|
| 3 habitation | 23 cabane   | 43 casse     | 63 cagna    | 83 mas       |
| 4 logis      | 24 gourbi   | 44 taudis    | 64 reposeé  | 84 tanière   |
| 5 domicile   | 25 gîte     | 45 chalet    | 65 chartruese | 85 cache     |
| 6 pénetes    | 26 berceau  | 46 pavillon  | 66 domesticité| 86 niche     |
| 7 home       | 27 masure   | 47 villa     | 67 garde-meubles| 87 rendez-vous_de_chasse|
| 8 mensil     | 28 aile     | 48 chaumiére | 68 gabionade| 88 repaire   |
| 9 chacunier  | 29 château  | 49 manse     | 69 lignée   | 89 clinique  |
| 10 foyer     | 30 immeuble | 50 hôtel_particulier| 70 caponnier| 90 gloriette|
| 11 train_de_maison | 31 hutte      | 51 isba     | 71 fermette | 91 havre    |
| 12 logement  | 32 maisonnette | 52 bas-lieu | 72 grand_ensemble| 92 lapinière|
| 13 maisonnée | 33 meul   | 53 manoir    | 73 lieu     | 93 kiosque  |
| 14 nid       | 34 abri    | 54 hôtel     | 74 famille  | 94 interieur|
| 15 résidence | 35 gazet    | 55 standing  | 75 garde-meubles| 95 bouverie|
| 16 toit      | 36 laves   | 56 H.L.M.    | 76 deck-house | 96 parents  |
| 17 bicoque   | 37 lare    | 57 habitacle | 77 habitat  | 97 tourelle |
| 18 bâtisse   | 38 clapier | 58 retraite  | 78 edifice  | 98 mantelet |
| 19 caisse    | 39 palais  | 59 carbet    | 79 ferme    | 99 hangar   |
| 20 baraque   | 40 train_de_vie | 60 séjour   | 80 tranchée-abri| 100 pigeonnier|
3.2. Similarity matrices and clusterization

It is also straightforward to build a similarity matrix (see table 2) and to use such matrices to group terms by clusters, i.e. lists of terms which do not all belong to the same clique, but which are closely related semantically. We use nearest-neighbor clustering in this work.

| Table 2 - Similarity matrix |
|--------------------------------|
| chacunière | 1.000 |
| mesnil | 0.665 | 1.000 |
| train_de_maison | 0.657 | 0.659 | 1.000 |
| maisonnée | 0.546 | 0.554 | 0.560 | 1.000 |
| demeure | 0.456 | 0.470 | 0.449 | 0.394 | 1.000 |
| habitation | 0.414 | 0.433 | 0.403 | 0.344 | 0.852 | 1.000 |
| maison | 0.661 | 0.669 | 0.653 | 0.596 | 0.843 | 0.820 | 1.000 |
| pénates | 0.481 | 0.496 | 0.481 | 0.417 | 0.837 | 0.605 | 0.762 | 1.000 |
| logement | 0.257 | 0.278 | 0.250 | 0.223 | 0.796 | 0.724 | 0.603 | 0.616 | 1.000 |
| domicile | 0.457 | 0.464 | 0.462 | 0.394 | 0.780 | 0.670 | 0.767 | 0.620 | 0.588 | 1.000 |
| logis | 0.507 | 0.514 | 0.506 | 0.447 | 0.795 | 0.666 | 0.810 | 0.706 | 0.623 | 0.892 | 1.000 |
| résidence | 0.305 | 0.310 | 0.308 | 0.262 | 0.727 | 0.609 | 0.576 | 0.511 | 0.554 | 0.840 | 0.627 | 1.000 |

In table 3, the headers are the members of the original cliques including maison, grouped in semantically homogeneous clusters, and the associated lists are terms similar with $\sigma > 0.25$ to the center of mass of their header. Terms in blue are from the original cliques, terms in gray are repeats from a previous cluster, and the others could reasonably be aggregated to their head cluster, especially at similarities above 0.35.

| Table 3 - Clusters around maison and their cohorts |
|---------------------------------|
| chacunière, mesnil, train_de_maison, maisonnée, demeure, habitation, pénates, maison, logement, domicile, logis, résidence |
| 0.73 maison | 0.55 chacunière | 0.49 foyer | 0.33 bercaill | 0.32 cahute | 0.28 lare |
| 0.70 demeure | 0.55 résidence | 0.47 maisonnée | 0.33 château | 0.32 ménil | 0.28 lées |
| 0.67 logis | 0.55 logement | 0.42 nid | 0.33 gîte | 0.30 asile | 0.26 cabane |
| 0.66 domicile | 0.54 mesnil | 0.42 ido | 0.33 biocoque | 0.30 masure |
| 0.63 pénates | 0.54 train_de_maison | 0.39 bâtisse | 0.33 casine | 0.29 baraque |
| 0.62 habitation | 0.49 home | 0.34 chez-soi | 0.33 gourbi | 0.29 immeuble |
| abri, clapier, gîte, nid, asile, retraite, bercaill, toit, foyer, habitacle |
| 0.54 abri | 0.41 clapier | 0.37 caponnière | 0.32 logis | 0.29 pare-éclats | 0.26 soue |
| 0.54 nid | 0.40 foyer | 0.36 repaire | 0.32 refuge | 0.29 domicile | 0.26 kiosque |
| 0.50 toit | 0.38 deck-house | 0.36 garde-meuble | 0.31 bouvierie | 0.29 home | 0.26 nique |
| 0.49 gîte | 0.38 retraite | 0.36 | 0.30 lapinière | 0.28 taud | 0.26 abrivent |
| 0.47 bercaill | 0.37 tranche-abri | 0.35 plate_de_blindage | 0.30 hangar | 0.27 reposée | 0.25 cagna |
| 0.46 asile | 0.37 habitation | 0.35 tanière | 0.30 mantelet | 0.27 reposoir | 0.25 porcherie |
| 0.43 maison | 0.37 gabionsade | 0.35 logement | 0.30 étable | 0.27 cachette | 0.25 tourelle |
| 0.43 demeure | 0.37 garde-meubles | 0.33 have | 0.29 lieu_sûr | 0.27 austen | 0.25 brise-vent |
| 0.41 pénates | 0.37 cache | 0.33 habitation | 0.29 tenderele | 0.27 darse |
| baraque, bicoque, cahute, cabane, hutte, gourbi, masure, case, casine, chaumière, maisonnette |
| 0.69 cabane | 0.59 hutte | 0.45 chaumière | 0.37 habitation | 0.30 buron | 0.25 chacunière |
| 0.65 baraque | 0.50 masure | 0.43 maison | 0.36 chaumine | 0.27 train_de_maison | 0.25 appellis |
| 0.64 bicoque | 0.50 maisonnnette | 0.42 case | 0.35 carbet | 0.26 chalet |
| 0.60 cahute | 0.50 gourbi | 0.40 casine | 0.32 cagna | 0.26 demeure |
| bas-lieu, naissance, origine, descendance, familie, lignée, race, parents, chez-soi, home, intérieur, ménil, lare, lares, ménage, standing, train_de_vie |
| 0.33 famille | 0.31 descendance | 0.27 maison | 0.25 bas-lieu |
| 0.33 race | 0.31 lignée | 0.26 fission |
| appartenent, bouge, tauds, galettes, chalet, pavillon, ville, château, manoir, palais, réduit |
| 0.33 château | 0.32 pavillon | 0.30 galettes | 0.29 maison | 0.27 manoir | 0.26 bouge |
| 0.33 habitation | 0.30 tauds | 0.29 villa | 0.29 chartreuse | 0.27 chalet |
| building, édifice, bâtiment, immeuble, construction, bâtisse, hôtel, campagne, propriété, ferme |
| 0.43 building | 0.40 bâtiment | 0.34 bâtisse | 0.30 H.L.M. |
| 0.41 immeuble | 0.36 édifice | 0.30 construction |
| boîte, entreprise, firme, établissement, prison, commerce, temple, institut, institution, branche, couvert, domesticié, serviteur, domestique, gens, monde, suite |
| clinique, hôpital, nom, couronne, trône, pigeonnier, lieu, place, séjour, feu |

3.3. Orthogonalization

Things get more complicated when two homonyms are semantically disjoint, as is the case with *le barde* and *la barde*:
If we meant *barde* as *aëde*, the third neighbor, *tranche_de_lard* is clearly not appropriate, and conversely. However, in a Euclidean space, the Schmidt orthogonalization procedure does remove this kind of interference. Since term vectors are normalized to unity, one needs simply to subtract from the vector \( \text{barde} \) the collinear component of the vector \( \text{tranche_de_lard} \):

\[
\|\text{barde}\rangle_{\perp \text{tranche_de_lard}} = \|\text{barde}\rangle - \langle\text{barde}\|\text{tranche_de_lard}\rangle \times \|\text{tranche_de_lard}\rangle
\]

with the following result, where the perturbation due to *tranche_de_lard* is totally eliminated:

| From 1 to 20 | From 21 to 40 | From 41 to 60 | From 61 to 80 | From 81 to 100 |
|-------------|--------------|--------------|--------------|---------------|
| 1 1.000     | 0.298        | 0.167        | 0.097        | 0.080         |
| 2 0.839     | 0.297        | 0.157        | 0.096        | 0.079         |
| 3 0.717     | 0.290        | 0.148        | 0.096        | 0.079         |
| 4 0.625     | 0.290        | 0.147        | 0.094        | 0.078         |
| 5 0.533     | 0.289        | 0.146        | 0.094        | 0.078         |
| 6 0.521     | 0.286        | 0.124        | 0.094        | 0.077         |
| 7 0.503     | 0.287        | 0.119        | 0.093        | 0.076         |
| 8 0.493     | 0.285        | 0.114        | 0.093        | 0.076         |
| 9 0.468     | 0.284        | 0.110        | 0.091        | 0.075         |
| 10 0.450    | 0.284        | 0.110        | 0.090        | 0.074         |
| 11 0.427    | 0.282        | 0.109        | 0.090        | 0.073         |
| 12 0.421    | 0.276        | 0.105        | 0.090        | 0.072         |
| 13 0.328    | 0.274        | 0.105        | 0.089        | 0.072         |
| 14 0.323    | 0.267        | 0.103        | 0.086        | 0.073         |
| 15 0.315    | 0.256        | 0.103        | 0.085        | 0.073         |
| 16 0.313    | 0.240        | 0.103        | 0.084        | 0.073         |
| 17 0.313    | 0.233        | 0.102        | 0.084        | 0.073         |
| 18 0.308    | 0.216        | 0.100        | 0.082        | 0.072         |
| 19 0.302    | 0.207        | 0.097        | 0.081        | 0.072         |
| 20 0.300    | 0.176        | 0.097        | 0.080        | 0.073         |

The number of terms which can be subtracted is only limited by the noise.
The corresponding clusters associated with \textit{barde} \_\textit{tranche_de_lard} now are:

| Table 6 - Clusters around \textit{barde} and their cohorts (orthogonalized w.r.t. \textit{tranche_de_lard}) |
|-----------------------------------------------|
| aède, barde, poète, chanteur, chantre |
| 0.71 chanteur                  | 0.55 troubadour                       | 0.44 fils_d’Apollon          | 0.43  | 0.42 favori_des_Muses          | 0.36 rapsode |
| 0.71 aède                     | 0.52 barde                            | 0.44 maître_du_Pinde         | 0.43  | 0.42 favori_des_Muses          | 0.36 rapsode |
| 0.70 scaldé                   | 0.48 rhapsode                         | nourrisson_du_Pinde         | 0.40  | 0.42 chœrute                  | 0.35 chœriste |
| 0.70 chantre                  | 0.47 duettiste                        | nourrisson_du_Pinde         | 0.40  | 0.42 chœrute                  | 0.35 chœriste |
| 0.66 poète                    | 0.46 crooner                          | nourrisson_du_Pinde         | 0.40  | 0.42 chœrute                  | 0.35 chœriste |
| 0.61 minnesinger              | 0.46 clithéréde                       | nourrisson_du_Pinde         | 0.40  | 0.42 chœrute                  | 0.35 chœriste |
| 0.55 trouvère                 | 0.44 poëtèrau                         | nourrisson_du_Parnasse      | 0.42  | 0.42 favori_d’Apollon         | 0.37 versificateur |
| bardit, harnais, prêtre, lamelle, panne, rhapsode, troubadour, trouvère |
| 0.29 trouvère                 | 0.28 minnesinger                      | nourrisson_du_Parnasse      | 0.36  | 0.27 coryphée                 | 0.28 vocaliste |
| 0.29 troubadour               | 0.28 minnesinger                      | nourrisson_du_Parnasse      | 0.36  | 0.27 coryphée                 | 0.28 vocaliste |

To be compared to the non-orthogonalized result:

| Table 7 - Clusters around \textit{barde} and their cohorts |
|-----------------------------------------------|
| aède, barde, chanteur, chantre, poète |
| 0.79 aède                                 | 0.54 trouvère                          | 0.40 fils_d’Apollon          | 0.40  | 0.39 favori_d’Apollon         | 0.32 choriste |
| 0.72 chantre                              | 0.52 troubadour                        | 0.40 maître_du_Pinde         | 0.40  | 0.39 favori_d’Apollon         | 0.32 choriste |
| 0.70 barde                                | 0.50 rhapsode                          | nourrisson_du_Parnasse      | 0.40  | 0.39 favori_d’Apollon         | 0.32 choriste |
| 0.69 chantre                              | 0.42 duettiste                         | nourrisson_du_Parnasse      | 0.40  | 0.39 favori_d’Apollon         | 0.32 choriste |
| 0.64 scaldé                               | 0.42 clithéréde                       | nourrisson_du_Parnasse      | 0.40  | 0.39 favori_d’Apollon         | 0.32 choriste |
| 0.63 poète                                | 0.42 crooner                          | nourrisson_du_Parnasse      | 0.40  | 0.39 favori_d’Apollon         | 0.32 choriste |
| 0.56 minnesinger                          | 0.41 poëtèrau                         | nourrisson_du_Parnasse      | 0.40  | 0.39 favori_d’Apollon         | 0.32 choriste |
| bardit, \textit{tranche_de_lard}, lamelle, rhapsode, troubadour, trouvère |
| 0.48 bardit                               | 0.41 aède                              | 0.34 rhapsode                | 0.31  | 0.28 bardit                   | 0.25 scaldé |
| 0.42 troubadour                           | 0.37 minnesinger                      | 0.33 \textit{tranche_de_lard} | 0.30  | 0.29 bardit                   | 0.25 scaldé |
| 0.41 troubadour                           | 0.35 poète                             | 0.32 ménestrel               | 0.29  | 0.29 bardit                   | 0.25 scaldé |
| harnais, panne, prêtre                    |
| 0.35 panne                                | 0.27 corybante                         | 0.27 flamme                 | 0.27  | 0.26 mystagogue               | 0.26 auffublement |
| 0.34 harnais                              | 0.27 mufi                              | 0.27 curéte                 | 0.27  | 0.26 mystagogue               | 0.26 auffublement |
| 0.34 prêtre                               | 0.27 iman                             | 0.27 hérogrammate            | 0.27  | 0.26 mystagogue               | 0.26 auffublement |
| 0.28 muezzin                              | 0.27 ovate                            | 0.27 archiprétre             | 0.27  | 0.26 mystagogue               | 0.26 auffublement |

### Conclusion and future work

We have shown how to construct a Euclidean semantic space from a French synonym dictionary by combining random vectors and random projection with the usual vector space model. The process is extremely fast and introduces an amount of noise acceptable in most situations. Updating the semantic space with new synonyms involves handling only a limited number of terms and is thus easily done in real time. As usual, the inner product between two normalized term vectors in the resulting semantic space is interpreted as the similarity between the terms. In additions, the Schmidt orthogonalization process is immediately applicable and can be used to separate semantically disjoint homonyms. The resulting semantic space is directly suited to the generation of practical lists of synonyms and to applications such as word selection for automatic text generation.

This process can be applied to any language, including English, where for example WordNet could be used as a basis. It can also be easily extended to collocations or to contextexons \cite{15,16,17}. In future work, we plan on expanding this method by building a dynamic semantic space from Wikipedia and its “history” pages, whereby the evolution of the meaning of terms as a function of time can be viewed through the evolution of their semantic neighborhoods. This could also be done using time series from newspapers, from radio and television. We are also considering building a real-time semantic space from news and social networking services such as Twitter.
Bibliography

[1] Ploux, S. and Victorri, B., Construction d'espaces sémantiques à l'aide de dictionnaires de synonymes, *Traitement Automatique des Langues, ATALA* (1998) Pages 161-182.

[2] Salton, G., Automatic Information Extraction and Retrieval, 1968.

[3] Salton, G., Automatic Text Processing, *Addison-Wesley Publishing Company*, ISBN 0-201-12227-8, 1989.

[4] Ji, H., J., Ploux, S., Wehrli, E., Lexical Knowledge Representation with Contextonyms, *Proceedings of the 9th MT summit*, 2008, pp. 194-201.

[5] Deerwester, S. et al., Improving Information Retrieval with Latent Semantic Indexing, *Proceedings of the 51st Annual Meeting of the American Society for Information Science* (25), 1988.

[6] Lebart, L. and Salem, A., Statistique Textuelle, p. 91, Dunod, 1994.

[7] Goldberg, Y., A Primer on Neural Network Models for Natural Language Processing, Draft, 2015.

[8] Mikolov, T., Chen, K., Corrado, G. and Dean, J., Efficient Estimation of Word Representations in Vector Space, arXiv:1301.3781v3, 2013.

[9] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean, J., Distributed Representations of Words and Phrases and their Compositionality, arXiv:1310.4546v1, 2013.

[10] Dasgupta S., Technical Perspective: Strange Effects in High Dimension, *Communications of the ACM, Vol. 53* (2010) No. 2, Page 96.

[11] Johnson, W. and Lindenstrauss, J., Extensions of Lipschitz maps into a Hilbert space, *Contemporary Mathematics 26*, (1984), pp. 189-206.

[12] Sahlgren, M., The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations between Words in High-dimensional Vector Spaces, *PhD Dissertation*, Stockholm University, Sweden, 2006..

[13] QasemiZadeh, B. and Handschuh, S., Random Indexing Explained with High Probability, *Proceedings of the 18th International Conference on Text, Speech and Dialog*, Springer International Publishing, 2015.

[14] Paradis, R., Guo, J.K., Moulton, J., Cameron, D. and Kanerva, P., Finding Semantic Equivalence of Text Using Random Index Vectors, *Procedia Computer Science* 20, 2013, pp. 454-459.

[15] Boussidan, A., Dynamics of semantic change, *PhD Dissertation*, Université Lumiére-Lyon2, France, 2013.

[16] Ji, H., J., Ploux, S., Wehrli, E., *ibid*.

[17] Boussidan, A., Renon A.-L., Franco C., Lupone S. and Ploux, S., Repérage automatique de la néologie sémantique en corpus à travers des représentations cartographiques évolutives, To be published, 2010?.

10/10