Lexical sorting and lexical transfer by conceptual vectors

Mathieu Lafourcade
LIRMM - UMR 5506, Université Montpellier II - CNRS
161, rue Ada. 34392 Montpellier Cedex 5, France
lafourca@lirmm.fr - http://www.lirmm.fr/~lafourca

Abstract

This paper introduces a conceptual representation based on concept vectors. These conceptual vectors aims at representing the set of ideas related to a given text segment. Vector operations associated with a classical morphosyntactical analysis can pave the way to an efficient lexical disambiguation. Lexical transfer can also be undertaken graciously with neither a transfer lexicon nor a pivot approach.

1 Introduction

In the framework of Natural Language Processing (NLP), we focus at the lexical sorting or disambiguation. Among other strategies, the processing of lexical selection can be statistical ([Salton and MacGill 1983], [Yarowsky 1995], [Yarowsky 1994]), without syntax and based on key words [Salton 1988], or based on taxonomies [Resnik 1995]. Approaches should avoid two pitfalls: dealing with representations either too poor to be widely efficient, either requiring a formidable workforce for indexation.

From a set of elementary concepts as found in thesaurii, it is possible to construct a vector space, and associate word meaning to vectors. Polysemic words are then related to several vectors, each of then associated to a meaning. This vectorial approach is built on well known mathematical basis and well founded manipulations with reasonable linguistic interpretations are then possible. We can this way introduce the notions of vector augmentation and contextualization and also define a distance function corresponding to the conceptual proximity. The *augmentation* iterative operation takes into account the meaning neighborhood of concepts to refine vectors. These vectors are then used for a *semantic* analysis which results (*contextualized* conceptual vectors) can lead to meaning sorting (and eventually disambiguation) and transfer.

2 Conceptual vectors

The definition of *conceptual vectors* aims at a thematic representation of textual segments independently of the considered syntactic level (words, phrases, sentences and texts). A vector is a projection of expressed ideas on a set of predetermined concepts which are extracted from conventional thesaurii.

2.1 Thesaurus and vector space

The basic idea behind conceptual vector is to consider a theme as a composition of basic concepts. Before going further the question of which concepts are likely to be selected arises. Thesaurii consitute adequate sources for concept set selection. The size of the concept set of a typical thesaurus is around 1000 - for example there are 882 concepts for [Carney and Wait 1985], 873 in [Larousse 1992] and more than 1000 in [Kirkpatrick 1987].

From this set of n concepts, we can define a vector space $\mathcal{E}$ of dimension p with $p \leq n$. The set of n vectors related to the n concepts is a generator set of $\mathcal{E}$. The choosen thesaurus is the sole source of information needed for building $\mathcal{E}$. A conceptual vector $V_{\mathcal{E}}$ that belongs to $\mathcal{E}$ is then is a p-uple. Each vector component corresponds to a concept of the thesaurus and is a linear combination of the generator set. A meaning can be represented by a vector $V_{\mathcal{E}}$ or by a vector V which is a linear combination of $V_{i\mathcal{E}}$. In our case, V is redundantly represented by a n-uple.

The semantic fields related to concepts are not independent and it may be difficult to differentiate them or to give a computable relation between them. For example, antonym relation is clear between notions like AGREEMENT$_C$ and DISAGREEMENT$_C$ and thus could lead to an opposition between these two dimensions in $\mathcal{E}$. Their coefficients could then be set to 1 or -1. We deliberately put aside this possibility because we consider the negation of a concept doesn't modify the semantic reference to this concept.

Although not needed, it is important to note that a base of $\mathcal{E}$ is not directly given by the thesaurus because the vectors associated to the concepts of the thesaurus are *not* independant.

A given dimension of a conceptual vector is mathematically defined in [0,1] of $\mathbb{R}$. This value represents the *intensity* of its related notion in the vector (0 no activation, 1 maximal unique activation). The vector norm is always equal to 1. From our implementation point of view, a vector is a n-uple of integers (of values between 0 and $2^{32}-1$).

2.2 Raw conceptual vectors

A raw vector of a word w (noted *r-vector(w)*) does only contain references to concept immediately related of w (as found in the index part of the thesaurus). The indexation process of r-vectors is essentially done by hand or can be computed from the main part of the thesaurus.

For example, in [Larousse 1992], the feminine headword FRICTION$_H$ is linked to the following notions: FROTTEMENT$_C$, SOINS DU CORPS$_C$, DÉSACCORD$_C$, CONFLIT$_C$ and COIFFURE$_C$ (which respectively correspond to dimension numbers 228, 393, 429, 649, 867). It can be translated as the following vector V^0:

⟨FROTTEMENT$_{1/\sqrt{5}}$, SOINS-DU-CORPS$_{1/\sqrt{5}}$, DÉSACCORD$_{1/\sqrt{5}}$, CONFLIT$_{1/\sqrt{5}}$, COIFFURE$_{1/\sqrt{5}}$⟩

The value $1/\sqrt{5}$ is the result of the normalization of the vector. All other notions have 0 as value. The building of a raw vector consists in setting each related notions to 1 and then to normalize. If a word is directly related to n notions, then each related dimension will have the value $1/\sqrt{n}$.

2.3 Augmented conceptual vectors

After a quick experimentation, it became clear that the use of r-vectors alone didn't lead to satisfactory results and was not conceptually different than [Salton 1988]. We identified the following reasons:

- Although, intuitively two notions may have at least an indirect relation (for instance SICKNESS$_C$ and MEDECINE$_C$), this relation may not be apparent in r-vectors.
- Conversely, if some notions are not expressed in a vector, unrelated related notions will be more likely to interfere during propagation (cf 3.2) leading to undesirable results.
- Finally, notions are hierarchically grouped in a thesaurus according an ontology. The notion group in which appears a given notion of a vector is naturally an immediate meaning neighborhood. For example, antonyms are strongly related and r-vectors do not always render properly this issue.

Taking into account the vector neighborhood consists, for each notion, in filling a vector with related concepts (in the spirit of what [Véronis and Ide 1990] applied on networks). We called *augmentation*, the process of computing a new *a-vector(w)$'$* from the neighborhood of *a-vector(w)*. For a headword w it is iteratively defined as follows:

$$V^{i+1}(w) = V^i(w) + \sum_{k=1}^{dim(V)} \alpha_i V^i_k(w) \times V^0(H_k)$$

- $V^i(w)$ is the *a-vector(w)* up to iteration i; by definition *r-vector(w)*$= V^0(w)$.
- $dim(V)$ is the dimension of the vector (the size of $\mathcal{E}$).
- α_i is a postive decreasing factor; we can choose for example $\alpha_i = 1/2^i$.
- H_k is the headword of the kth concept; for example, in [Carney and Wait 1985] we have H_{333} =FEMALE$_H$.
- $V_k(w)$ is the kth dimension of $V(w)$.

When $\lim_{i \to \infty} \alpha_i = 0$, the augmentation process is convergent (respectively to the *dist* function defined in 2.5).

For instance, the word FRICTION$_H$, we have the following a-vectors (for the sake of clarity, their norm has been made equal to 1000 and only non null components are present):

V^0 : ⟨FROTTEMENT$_{447}$, SOINS-DU-CORPS$_{447}$, DÉSACCORD$_{447}$, CONFLIT$_{447}$, COIFFURE$_{447}$⟩

V^1 : ⟨FROTTEMENT$_{472}$, CONFLIT$_{444}$, DÉSACCORD$_{365}$, SOINS-DU-CORPS$_{349}$, COIFFURE$_{344}$, GUERRE$_{187}$, ÉNERGIE$_{180}$, MÉCANIQUE$_{177}$, MUSIQUE$_{173}$, TRIBUNAL$_{167}$, INIMITIÉ$_{138}$, POLI$_{127}$, ADVERSITÉ$_{116}$, DESSUS$_{101}$, MACHINE$_{65}$, RÉVOLUTION$_{64}$, DÉFIANCE$_{57}$, MOUVEMENT$_{54}$, CONSEIL$_{50}$, GUÉRISON$_{46}$, DIVORCE$_{45}$ $\cdots$⟩

V^2 : ⟨CONFLIT$_{387}$, DÉSACCORD$_{312}$, GUERRE$_{296}$, FROTTEMENT$_{278}$, SOIN-DU-CORPS$_{258}$, TRIBUNAL$_{254}$, COIFFURE$_{249}$, ÉNERGIE$_{221}$, MÉCANIQUE$_{219}$, MUSIQUE$_{214}$, INIMITIÉ$_{201}$, ADVERSITÉ$_{197}$, DESSUS$_{150}$, MACHINE$_{138}$, RÉVOLUTION$_{134}$, MOUVEMENT$_{123}$, POLI$_{118}$, POLITIQUE$_{116}$, CONSEIL$_{116}$, MANOEUVRE$_{104}$, PASSE-TEMPS$_{99}$, FORCE$_{90}$, DÉFIANCE$_{87}$, DÉTECTION$_{82}$, ENSEIGNEMENT$_{75}$, GUÉRISON$_{73}$, DISCORDANCE$_{73}$, DIVORCE$_{70}$, TRANSPORT-PAR-ROUTE$_{69}$, SON$_{68}$, MÉDECINE$_{66}$, DISSEMBLANCE$_{64}$, PAIX$_{61}$, PROFANE$_{60}$, JEU$_{53}$, MALADIE$_{51}$, VIGUEUR$_{47}$, DIFFICULTÉ$_{46}$, MAIN-D'OEUVRE$_{43}$, CHOC$_{43}$, REPOS$_{38}$, IMMUNITÉ$_{38}$ $\cdots$⟩

2.4 Contextualized conceptual vectors

Contextualized vectors (aka *c-vectors*) are those resulting from a propagation process done over $\mathcal{T}$, the tree result of a morphosyntactic analysis. This *conceptual analysis* is described in (cf 3.2). At this stage, the vector related to each word has been *modified by its context* in $\mathcal{T}$.

2.5 Semantic Lexicon

We call *semantic lexicon headword* a 3-uple ⟨*string, set of morphological features (as cat, gender, etc.), meaning#*⟩. For example, from [Larousse 1992], the indexing of *feu* gives 8 nouns and 2 adjectives:

feu$_{(N,M)-1}$ as *incandescence*,
feu$_{(N,M)-2}$ as *lumière*,
feu$_{(N,M)-3}$ as *famille*,
feu$_{(N,M)-4}$ as *cheminée*,
feu$_{(N,M)-5}$ as *brûlure*,
feu$_{(N,M)-6}$ as *rougeur*,
feu$_{(N,M)-7}$ as *enthousiasme*,
feu$_{(N,M)-8}$ as *arme*,
feu$_{(ADJ,M)-1}$ as *mort*, and
feu$_{(ADJ,M)-2}$ as *rouge*.

In [Carney and Wait 1985], the index for *fire* is simpler (and not polysemic!):
fire$_{(N)}$ and
fire$_{(VB)}$.

A semantic lexicon $\mathcal{L}$ is a set of ⟨headword, vector⟩ that allows three operations :

Semantic Vector Search $SV_L(w)$: from the headword w, to find its conceptual vector V, if $\langle w, V \rangle \in \mathcal{L}$

Headword Search $SV_L(w)$: from the vector V, to find its headword w, if $\langle w, V \rangle \in \mathcal{L}$

k Nearest Neighbor Search $kNN_L(w)$: from a conceptual vector V, to find the k closest vector to V. Let $\mathcal{V}_k(v)$ be the neighborhood containing these k closest vectors and *dist* a distance function. The $\mathcal{V}_k(v)$ is defined as:

$$|\mathcal{V}_k(v)| = k$$
$$\forall V' \in \mathcal{V}_k(v) \quad \forall V'' \notin \mathcal{V}_k(v) \quad dist(V', V) \leq dist(V'', V)$$

How to define a proper *dist* function? In our context, we have to choose a function that is related to the semantic proximity of concepts. Intuitively, it can be defined as the cosine of the angle between two vectors. When two entities (words, phrases, texts) are semantically close, then their conceptual vectors should be proportionaly close. When colinear, two vectors represent with the same distribution of ideas. But, cos is not a mathematical distance function (because $\cos(x, x) \neq 0$).

In $\mathcal{E}$, for any vector V, we force $|V| = 1$ and $V_i \geq 0$. Thus, $0 \leq \widehat{V_1, V_2} \leq \pi/2$. The arccos function is a good candidate for *dist*, which is then defined as follows:

$$\forall V_1, V_2 \in \mathcal{E} \quad dist(V_1, V_2) = \arccos(V_1, V_2)$$

Other distance functions might have been possible (like the euclidian norm, the Mahattan or Chebychev distances to mention a few) although only arccos seems to fit a reasonable interpretation for our purpose.

3 Lexical sorting

3.1 Vector Functions

An analysis of a given text produces a syntactic tree $\mathcal{T}$ which contains classical morphosyntactic information. To predefined syntactic schemas S, we associate a vector function F. Although, it is out of the scope of this paper to present in detail each function, here are some examples :

- simple Noun phrase :
  $S : NP(ART, N) \to V(NP) = V(N)$.

- compound Noun phrase :
  $S : NP_1(NP_2, N) \to V(NP_1) = \alpha V(NP_2) + V(N)$ with $0 \leq \alpha \leq 1$. Usually, we choose $\alpha = 1/2$.

  For example $V(sail\ boat) = 1/2V(sail) + V(boat)$ and $V(boat\ sail) = 1/2V(boat) + V(sail)$.

- Abverb :
  $S : GA(GADV(ADV), ADJ) \to V(GA) = V(ADJ)^{p(ADV)}$.

  For example $p(very) = 2$ and $p(not\ very) = 1/2$. In this case, the negation of an adverb is its inverse function.

3.2 Vector Propagation Algorithm

The *c-vectors* are those present in nodes of $\mathcal{T}$ after propagation. The main steps of the vector propagation algorithm are the following:

1. on each node of $\mathcal{T}$ related to a contents word w of the text (i.e. with $cat(w) \in \{N, V, ADJ, ADV\}$) **attach** its a-vector found in $\mathcal{L}$. When a polysemic headword w has n meanings indexed as headword w_i with $1 \leq i \leq n$, then the a-vector of w is the mean of all submeaning a-vectors:

$$V(w) = \frac{1}{n} \sum_{i=1}^{n} V(w_i)$$

2. **Propagate** vectors along $\mathcal{T}$ up to the root. Let a node N_i be the root of a syntactic schema S with k dependants N_{i_j}, then:

$$V(N_i) = F_S(N_{i_1}, \cdots, N_{i_j}, \cdots, N_{i_k})$$

3. **Back-propagate** vectors along $\mathcal{T}$ down to the leaves. Let a node N_i and its k dependants N_{i_j}, then:

$$V(N_{i_j}) = V(N_{i_j}) \otimes V(N_i)$$

   where $\otimes$ is the term to term product.

3.3 lexical sorting and selection

For a headword w with n meanings w_i with a c-vector(w) in the context of $\mathcal{T}$, sorting the n meanings w_i for the most to the least probable, consists in sorting in increasing order the set $\{dist(\text{c-vector}(w), \text{a-vector}(w_i))\}$ with $1 \leq i \leq n$.

The *local selection* of the most probable meaning consists in fetching the *1NN*$_{\mathcal{S}}$(c-vector(w)) with $\mathcal{S} = \{\langle w_i, \text{a-vector}(w_i)\rangle\}$. If needed, the *global selection* of the most probable meaning consists in fetching the *1NN*$_{\mathcal{L}}$(c-vector(w)).

4 Lexical transfer

The lexical transfer aims at choosing the most appropriate headword contained in the target language lexicon $\mathcal{L_T}$ from a headword in the source language lexicon $\mathcal{L_S}$. Our approach allows us to process this stage through vectors.

4.1 Transfer Matrix

We define (w_S, w_T) as the association of a headword $w_S \in \mathcal{L_S}$ and its closest equivalent $w_T \in \mathcal{L_T}$. We also define a *transformation matrix* $\mathcal{M}$ that transforms a vector V in another vector V'. $V' = \mathcal{M}(V)$ is called the *equivalent of V by* $\mathcal{M}(V)$. A *transfer matrix* $\mathcal{M}_{\mathcal{S}/\mathcal{T}}$ is a transformation matrix with the following properties:

$$\forall (w_S, w_T) \quad dist(\mathcal{M}_{S/T}(SV_S(w_S)), SV_T(w_T)) \leq t$$

Of course, the smaller the threshold t, the better. We empirically found $t = \pi/8$ a good compromize between accuracy and overconstraints.

For example, if we consider that *fire* is the English equivalent of the French word *feu*, then we should have $\mathcal{M}_{French/English}(SV(\textit{feu}))$ reasonably close to $SV(\textit{fire})$.

The building of a matrix $\mathcal{M}$ is currently done by hand (although there might be interesting possibilities in automating this process). We approximate $\mathcal{M}$ as reasonably good when it does fulfill the above condition for a large portion of a bilingual lexicon. We may of course have several matrixes corresponding to different knowledge fields, which is easier to achieve than a unique general one.

4.2 Transfer with indirect or direct selection

Let V_w be the c-vector($w_\mathcal{S}$) in $\mathcal{T}$ for a headword w. In the *indirect selection* strategy, we select first the closest vector V'_w of V_w in $\mathcal{L}_\mathcal{S}$ and look for the equivalent of V'_w by $\mathcal{M}_{\mathcal{S}/\mathcal{T}}$ in $\mathcal{L}_\mathcal{T}$. In the *direct selection* strategy, we directly look for the equivalent of V_w by $\mathcal{M}_{\mathcal{S}/\mathcal{T}}$ in $\mathcal{L}_\mathcal{T}$.

- **indirect selection**: Let $V_1 = \mathit{1NN}_{\mathcal{L}_\mathcal{S}}(V_w)$ be the closest vector in $\mathcal{L}_\mathcal{S}$ and let $V_2 = \mathcal{M}_{\mathcal{S}/\mathcal{T}}(V_1)$. Then, we retrieve the target word :

$$w_{\mathcal{L}_\mathcal{T}} = \mathit{1NN}_{\mathcal{L}_\mathcal{T}}(V_2).$$

- **direct selection**: Let $V_1 = \mathcal{M}_{\mathcal{S}/\mathcal{T}}(V_w)$. Then, we retrieve the target word :

$$w_{\mathcal{L}_\mathcal{T}} = \mathit{1NN}_{\mathcal{L}_\mathcal{T}}(V_1).$$

We found much better results when using a transfer with *direct* instead of *indirect* selection. The c-vector($w_\mathcal{S}$) is a fine description of the meaning of w in the context of $\mathcal{T}$, and as the *direct* selection first goes through a projection over $\mathcal{L}_\mathcal{S}$, the amount of information is impoverished before the transfer. Projecting the $w_\mathcal{S}$ meaning directly over $\mathcal{L}_\mathcal{T}$ seems to be more faithfull to the transfer process essence.

5 Discussion and Conclusion

The delicate part of our approach is not to invent new mathematical vector operations, but to choose among the existing ones, those wo are suitable. For example, is *arccosine* the most approriate distance function or is the backpropagation function sound?

The philosophy behing our approach is to sort meanings and to select a meaning as late as possible. The selected a-vector of a polysemic word has not necessarily to be close to the c-vector to be valid. It should only be the closest. Experiments showed very often, that just a subtle vector shift is needed for an appropriate meaning sorting.

The larger the dimension of $\mathcal{E}$, the more detailled a vector. Each word could be a concept and then the approach is similar to (Véronis and Ide 1990). Conversely, the dimension of $\mathcal{E}$ has a negative impact on computation time.

We are still in the process of building our semantic lexicons for French and English and evaluating our approach both for lexical sorting and lexical transfer. We have conducted tests on a reduced set of texts which sentences contains typically ambiguous terms. In the current state of the indexation of our r-vectors, the polysemic term meanings are appropriately sorted in about 90% of the cases, which is quite encouraging. The approach is not ampered by numerous (more than 2) submeanings and as we are using analysis tree (and not the text itself) for context lookup, long range dependancies is not a limiting factor.

We also developed a text indexer (for French) and its associated search engine, and testbed them on a set of roughly 2200 articles of the french newspaper Le Monde (accessible at `http://www.lirmm.fr/~lafourca`). Altough a full presentation and evaluation of this system is beyond the scope of this paper, the recall and precision are subtantically better than with word based indexers and search engines. Both close and loosely associated articles are recalled with preferences with strong thematic and somehow rhematic similarities. These results hold only with requests which scope is thematic (common nom, verbs) with associated vectors properly indexed in lexicons. Request based on proper nouns or otherwise associated to words not present in lexicon are deemed to fail, contrary to word based approaches.

References

[1] Kirkpatrick B. *Rogets Thesaurus of English Words and Phrases.* Penguin Books, 1986.

[2] Lafourcade M. and E. Sandford *Analyse et désambiguïsation lexicale par vecteurs*

sémantiques. In Proc. of TALN'99 (Cargèse, July 1999) pp 351-356.

[3] Larousse. *Thésaurus Larousse des idées aux mots des mots au idées.* Larousse, ISBN 2-03-320-148-1, 1992.

[4] Carney, F. and M. Wait *The Penguin Pocket English Thesaurus.* Penguin Books, ISBN 0-14-051193-8, 1985.

[5] Resnik P. *Using Information contents to evaluate semantic similarity in a taxonomy.* In Proceedings of IJCAI-95, 1995.

[6] Salton G. *Term-Weighting Approaches in Automatic Text Retrieval.* McGraw-Hill computer science serie. McGraw-Hill, Volume 24, 1988.

[7] Sparck Jones K. *Synonymy and Semantic Classification.* Edinburgh Information Technology Serie, 1986.

[8] Véronis J. and N. Ide *Word Sence Disambiguation with Very Large Neural Network Extracted from Machine Readable Dictionaries.* In Proc. 13th ICCL, COLING'90. (Helsinki, Finland, july 1990) vol 2/2, pp 389-394.

[9] Yarowsky, D. *Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French.* In Proc. of the Conference of ACL. (Las Cruces, NM., June 27-30, 1994.) Morgan Kaufmann, San-Francisco, pp 88-95.

[10] Yarowsky, D. *UNsupervised word sense diambiguation rivaling supervised methods* In Proc. of the ACL. (Cambridge, MA, June 26-30, 1995) Morgan Kaufmann, San-Francisco, pp 189-196.