Sémantique distributionnelle, embeddings (et dong)

felipe@iro.umontreal.ca

RALI
Dept. Informatique et Recherche Opérationnelle
Université de **Montréal**

V0.5

Last compiled: 11 novembre 2019

Plan

Approche distributionnelle

Before Deep : modèle vectoriel

And then came the “Deep”

- Word2Vec
- Analogie
- Meta-embeddings
- Évaluation
- Idées intéressantes
- Le cas bilingue

Évaluation

Plongements contextuels

rali

Plan

Approche distributionnelle

Before Deep : modèle vectoriel

And then came the “Deep”

- Word2Vec
- Analogie
- Meta-embeddings
- Évaluation
- Idées intéressantes
- Le cas bilingue

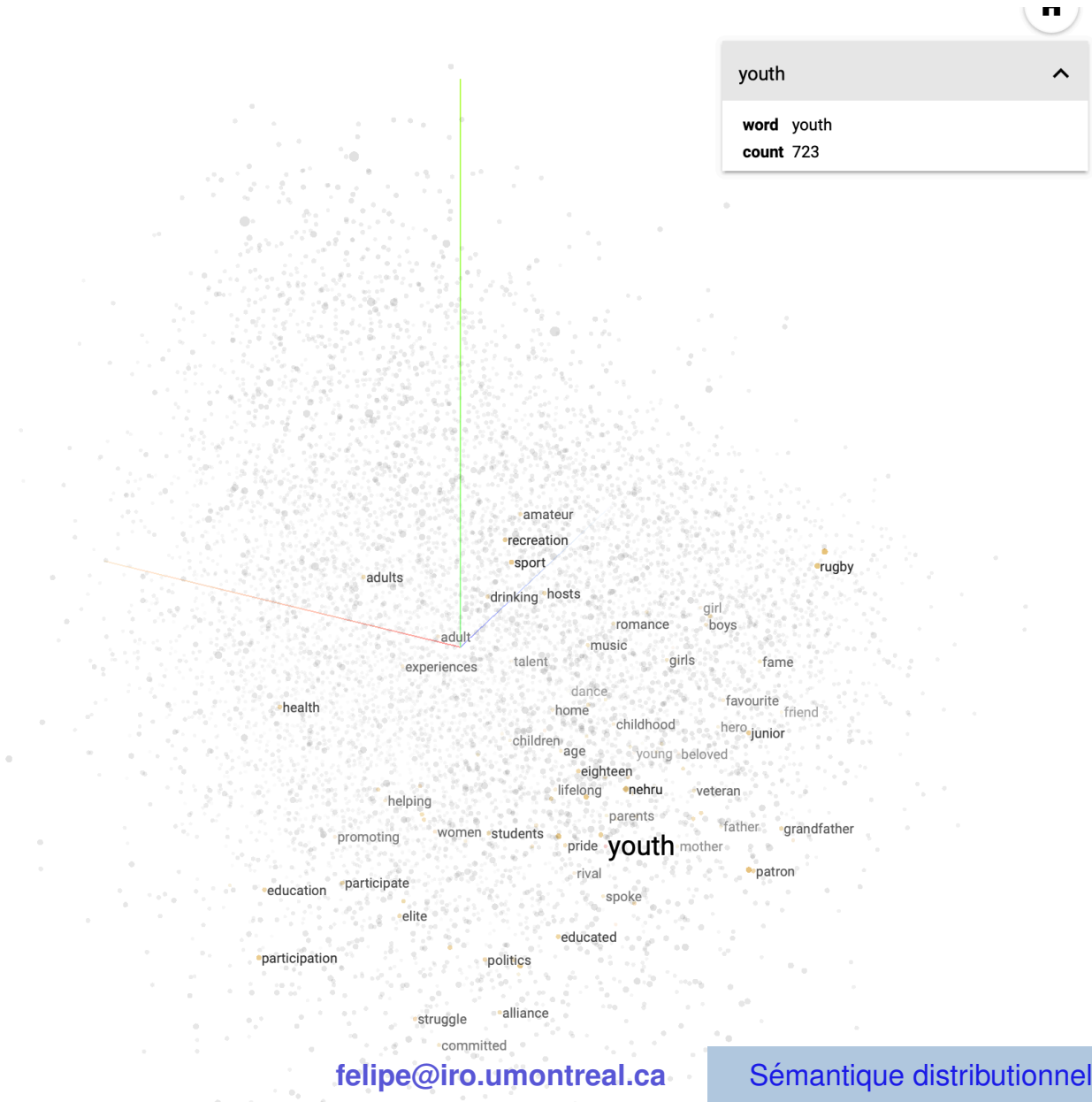
Évaluation

Plongements contextuels

rali

- If A and B have almost identical environments. . . we say that they are synonyms (Harris, 1954)
- you shall know a word by the company it keeps ! (Firth, 1957)
- words which are similar in meaning occur in similar contexts (Rubenstein & Goodenough, 1965)
- . . . In other words, difference of meaning correlates with difference of distribution (Harris, 1970, p.786)
- words with similar meanings will occur with similar neighbors if enough text material is available (Schütze & Pedersen, 1995)
- a representation that captures much of how words are used in natural context will capture much of what we mean by meaning (Landauer & Dumais, 1997)
- in the proposed model, it will so generalize because “similar” words are expected to have a similar feature vector, and because the probability function is a smooth function of these feature values, a small change in the features will induce a small change in the probability. (Bengio et al, 2003)

https://projector.tensorflow.org



Word2Vec entraîné sur un corpus de 510k blogs

email	76	email_address via_email mails text_messages sms snail_mail inbox letter error_message gmail_account notification invitation cv address
students	76	professors educators schools faculty grad_students college_students profs classrooms librarians high_schoolers administrators youngsters lecturers law_students
white	76	colored tights silk silky linen satin chunky plastic adorned dotted fur pasty thick fluffy
project	75	assignment research_project presentation term_paper assignments portfolio thesis collaborative proposal revisions coursework worksheet module presentations

Plan

Approche distributionnelle

Before Deep : modèle vectoriel

And then came the “Deep”
- Word2Vec
- Analogie
- Meta-embeddings
- Évaluation
- Idées intéressantes
- Le cas bilingue

Évaluation

Plongements contextuels

rali

Modèle vectoriel (Vector Space model)

- lire ***[Turney and Pantel, 2010]*** pour une introduction
- lire ***[Baroni and Lenci, 2010]*** pour une généralisation (tenseur)

  1. une matrice de “comptes” de co-occurences
  2. un schéma de pondération (PMI, LLR, etc.)
  3. une politique de réduction de dimensionnalité
     - *singular value decomposition* ***[Golub and Van Loan, 1996]***
     - *non-negative matrix factorization* ***[Lee and Seung, 1999]***
     - aucune ! (très bon *baseline*)
     - etc.

- DISSECT offre les étapes 2 et 3

rali

matrice de co-occurence : terme×document

- similarité de documents
- hypothèse bag of word : si une requête et un document ont des représentations (colonnes) similaires, alors ils véhiculent la même information *[Salton, 1975]*

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	5	117	0	0

Figure 19.1 The term-document matrix for four words in four Shakespeare plays. Each cell contains the number of times the (row) word occurs in the (column) document

- implémenté (par exemple) dans Lucene

. Pris de *[Jurafsky and Martin, 2015]*

matrice de co-occurence : terme×document

```
1: procedure TermxDoc(docs) ▷ set of tokenized documents
2:     V ← tr ' ' '\n' docs | sort | uniq
3:     for d ∈ docs do
4:         dict ← {}
5:         for w ∈ d do
6:             dict(w) ++
7:         end for
8:         print [dict(v) for v in V]
9:     end for
10: end procedure
```

matrice de co-occurence : terme×terme

- similarité de termes
- hypothèse distributionnelle : si deux termes ont des représentations (lignes) similaires, alors ils sont similaires

	aardvark	...	computer	data	pinch	result	sugar	...
apricot	0	...	0	0	1	0	1	
pineapple	0	...	0	0	1	0	1	
digital	0	...	2	1	0	1	0	
information	0	...	1	6	0	4	0	

Figure 19.4 Co-occurrence vectors for four words, computed from the Brown corpus, showing only six of the dimensions (hand-picked for pedagogical purposes). The vector for the word *digital* is outlined in red. Note that a real vector would have vastly more dimensions and thus be much sparser.

a

a. Pris de ***[Jurafsky and Martin, 2015]***

matrice de co-occurence : terme×rel

- similarité de termes
- hypothèse distributionnelle : si deux termes ont des représentations (lignes) similaires, alors ils sont similaires

	subj-of, absorb	*subj-of*, adapt	*subj-of*, behave	...	*pobj-of*, inside	*pobj-of*, into	...	*nmod-of*, abnormality	*nmod-of*, anemia	*nmod-of*, architecture	...	*obj-of*, attack	*obj-of*, call	*obj-of*, come from	*obj-of*, decorate	...	*nmod*, bacteria	*nmod*, body	*nmod*, bone marrow
cell	1	1	1		16	30		3	8	1		6	11	3	2		3	2	2

Figure 19.13 Co-occurrence vector for the word *cell*, from Lin (1998), showing grammatical function (dependency) features. Values for each attribute are frequency counts from a 64-million word corpus, parsed by an early version of MINIPAR.

a

a. Pris de ***[Jurafsky and Martin, 2015]***

matrice de co-occurence : terme×document

```
1: procedure TermxTerm(d)                                  ▷ d document
2:     V ← tr ' ' '\n' d | sort | uniq
3:     for i, w ∈ enumerate(d) do
4:         for c ∈ context(i, d) do
5:             rep[w][c] ++
6:         end for
7:     end for
8:     for v ∈ V do
9:         print rep[v]
10:    end for
11: end procedure
```

et par exemple : context(i,d) = $\{d[i-1], d[i-2], d[i+1], d[i+2]\}$

matrice de co-occurence : termes×terme

- similarité de relations
- hypothèse : si deux paires de mots ont des représentations (lignes) similaires, alors elles sont similaires *X of Y*, *Y of X*, *X for Y*, *Y for X*, *X to Y*, et *Y to X*
  - une liste de 64 mots comme *of*, *for* ou *to*
  - formant 128 patrons (colonnes) contenant la paire (X,Y) : ***[Turney, 2005]***

X	Y	X of Y	Y used by X	X for Y	Y for X
mason	stone	0	3	0	3
carpenter	wood	0	7	0	2

Plan

rali

Au menu

- un modèle vedette : Word2Vec *[Mikolov et al., 2013a]*
- propriétés des embeddings :
  [Mikolov et al., 2013d, Mikolov et al., 2013c]
- des résultats : glory *[Baroni et al., 2014]*, modération
  [Levy et al., 2015]
- cool works : *[Faruqui and Dyer, 2015, Faruqui et al., 2015b, Faruqui et al., 2015a]*
- modèles bilingues :
  [Mikolov et al., 2013b, Chandar et al., 2014, Gouws et al., 2015, Coulmance et al., 2016]

rali

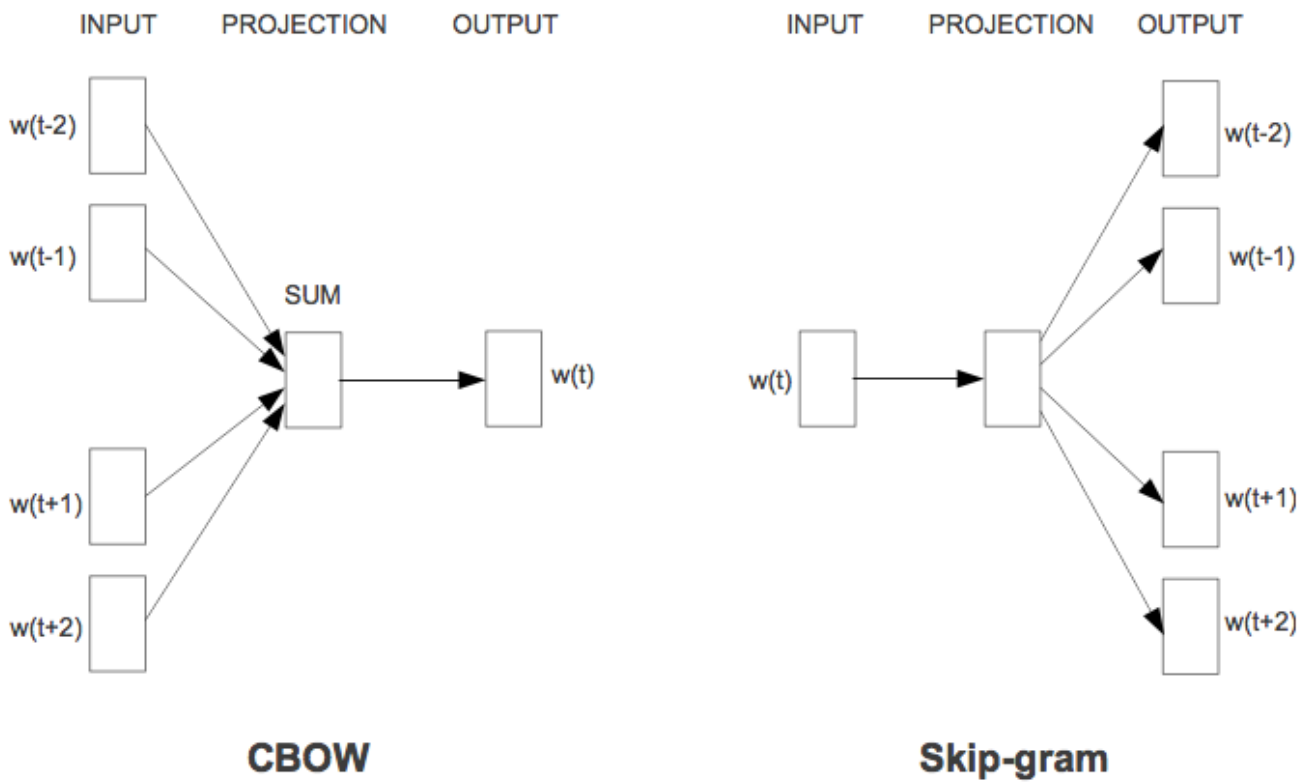
Une révolution chez les “distributionnalistes” : Word2Vec *[Mikolov et al., 2013a]*

- un toolkit rapide implémentant deux modèles
  - https://code.google.com/archive/p/word2vec/
  - https://radimrehurek.com/gensim/models/word2vec.html
  - https://github.com/dav/word2vec
- des embeddings disponibles entraînés sur 6B de mots de Google News (180K mots) - dimension = 300
- directement utilisable dans de nombreuses applications

rali

Les 2 modèles de Word2Vec

[Mikolov et al., 2013a]



- Skip-gram est le plus populaire (plus fiable pour les “petits” corpus)
- CBOW est plus rapide (bien pour les grands corpus)

Skip-gram *[Mikolov et al., 2013a]*

- C un corpus d'entraînement, aka un ensemble D de paires (w, c) où w est un mot de C et c est un mot vu dans un contexte note : le modèle représente différemment les mots de contexte des mots du vocabulaire
- Soit (w, c) ; est-ce une paire du corpus ?
  - $p(D = 1|w, c; \theta)$ la probabilité associée
- Optimise par descente de gradient :

$$L = \underset{\theta}{\operatorname{argmax}} \prod_{(w,c)\in D} p(D = 1|w, c; \theta) \prod_{(w,c)\in D'} 1 - p(D = 1|w, c; \theta)$$

  - D' est construit en choisissant k paires aléatoirement.

rali

Skip-gram *[Mikolov et al., 2013a]*

- $p(D = 1|w, c; \theta) = \sigma(v_c.v_w)$, alors :

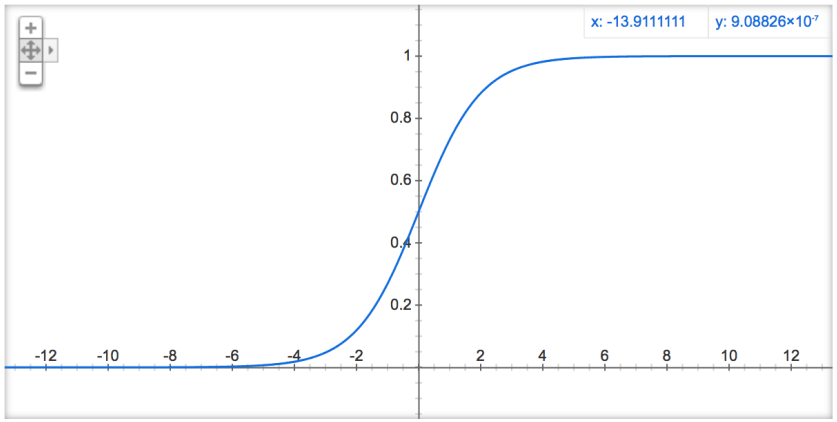
$$L = \underset{\theta}{\text{argmax}} \sum_{(w,c)\in D} \log \sigma(v_c.v_w) + \sum_{(w,c)\in D'} \log \sigma(-v_c.v_w)$$

  - $\sigma(x) = \frac{1}{1+e^{-x}}$,
  - v_c (resp. v_w) est le vecteur de c (resp. w)

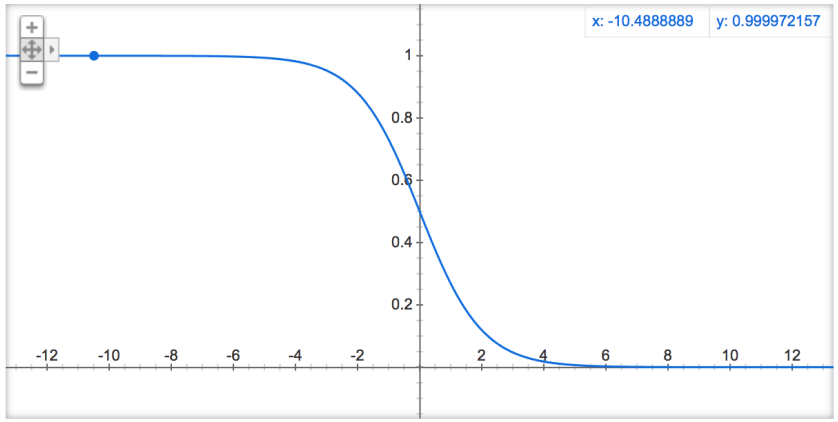
- les contextes sont définis par une fenêtre centrée autours du mot w considéré, et dont la taille est tirée aléatoirement (et uniformément sur un intervalle fixé)
- les mots les plus fréquents sont sous-échantillonnés (retirés aléatoirement de C) et les mots peu fréquents sont éliminés (cut-off).
- ça marche !!!
  - lire *[Levy and Goldberg, 2014]*

Skip-gram *[Mikolov et al., 2013a]*

Graphique pour 1/(1+exp(-x))



Graphique pour 1/(1+exp(x))



$v_c.v_w$ est :

- positif et d'autant plus grand que les vecteurs sont proches
  - premier terme maximisé
- négatif et d'autant plus petit que les vecteurs sont éloignés
  - second terme proche de 0

rali

Autres embeddings pré-entraînés

- Glove ***[Pennington et al., 2014]***

  glove.6B.zip (Wikipedia+GigaWord 2014, $|V|$=400K, $d \in \{50, 100, 200, 300\}$, 822Mo)

  glove42B.300d.zip (Common Crawl, $|V|$=1.9M, uncased, $d = 300$, 1.75 Go)

  glove.840B.300d.zip (Common Crawl, $|V|$=2.2M, cased, $d = 300$, 2.03 Go)

  glove.twitter.27B.zip (2B tweets, $|V|$=1.2M, uncased, $d \in \{25, 50, 100, 200\}$, 1.42 Go)

- word-embeddings-conll17.tar
  - pré-entraînés pour CONLL2017 à l'aide de Word2Vec
  - d=100, 45 langues, 630Go

rali

Embeddings multilingues

- Polyglot ***[Al-Rfou et al., 2013]***
  - 100 langues ! (Wikipedia)
  - entraîné à scorer des *phrases* du corpus mieux que des *phrases* dans lesquelles ont a remplacé un mot
- FastText ***[Bojanowski et al., 2016]***
  - 294 langues ! (Wikipedia)
  - skip-gram ou les mots sont représentés par des sacs de n-grams (caractère). Un embedding pour un mot inconnu peut donc être calculé.
- LASER ***[Schwenk et al., 2019]***
  - 21 langues, $d = 384$
  - 5 layers of BiLSTM (shared among languages)
  - 40k BPE (multilingual)

rali

Arithmétique analogique des représentations

[Mikolov et al., 2013d]

- vec(*Madrid*) - vec(*Spain*) $\simeq$ vec(*Paris*) - vec(*France*)
- permet de résoudre des équations analogiques : [x: y :: z :?]
  1. calculer $t = vec(y) - vec(x) + vec(z)$ le vecteur cible
  2. rechercher dans V, le mot $\hat{t}$ le plus proche de t :

$$\hat{t} = \underset{w}{\text{argmax}} \frac{vec(w).vec(t)}{||vec(w)|| \times ||vec(t)||}$$

[Mikolov et al., 2013d]

- RNN entraîné sur 320M de mots (V = 82k)

Method	Adjectives	Nouns	Verbs	All
LSA-80	9.2	11.1	17.4	12.8
LSA-320	11.3	18.1	20.7	16.5
LSA-640	9.6	10.1	13.8	11.3
RNN-80	9.3	5.2	30.4	16.2
RNN-320	18.2	19.0	45.0	28.5
RNN-640	21.0	25.2	54.8	34.7
RNN-1600	**23.9**	**29.2**	**62.2**	**39.6**

Table 2: Results for identifying syntactic regularities for different word representations. Percent correct.

- test set de 8k analogies impliquant les mots les plus fréquents

[Mikolov et al., 2013c]

- 6B de mots de Google News, 1M de mots les plus fréquents

Table 3: *Comparison of architectures using models trained on the same data, with 640-dimensional word vectors. The accuracies are reported on our Semantic-Syntactic Word Relationship test set, and on the syntactic relationship test set of [20]*

Model Architecture	Semantic-Syntactic Word Relationship test set		MSR Word Relatedness Test Set [20]
	Semantic Accuracy [%]	Syntactic Accuracy [%]	
RNNLM	9	36	35
NNLM	23	53	47
CBOW	24	64	61
Skip-gram	55	59	56

- le test syntaxique est le même que dans *[Mikolov et al., 2013d]*

rali

[Mikolov et al., 2013c]

- Comparaison à d'autres modèles proposés

Table 4: *Comparison of publicly available word vectors on the Semantic-Syntactic Word Relationship test set, and word vectors from our models. Full vocabularies are used.*

Model	Vector Dimensionality	Training words	Accuracy [%]		
			Semantic	Syntactic	Total
Collobert-Weston NNLM	50	660M	9.3	12.3	11.0
Turian NNLM	50	37M	1.4	2.6	2.1
Turian NNLM	200	37M	1.4	2.2	1.8
Mnih NNLM	50	37M	1.8	9.1	5.8
Mnih NNLM	100	37M	3.3	13.2	8.8
Mikolov RNNLM	80	320M	4.9	18.4	12.7
Mikolov RNNLM	640	320M	8.6	36.5	24.6
Huang NNLM	50	990M	13.3	11.6	12.3
Our NNLM	20	6B	12.9	26.4	20.3
Our NNLM	50	6B	27.9	55.8	43.2
Our NNLM	100	6B	34.2	**64.5**	50.8
CBOW	300	783M	15.5	53.1	36.1
Skip-gram	300	783M	**50.0**	55.9	**53.3**

rali

[Mikolov et al., 2013c]

- Big Data (plus de données, dimension plus élevée)

Table 6: *Comparison of models trained using the DistBelief distributed framework. Note that training of NNLM with 1000-dimensional vectors would take too long to complete.*

Model	Vector Dimensionality	Training words	Accuracy [%]			Training time [days x CPU cores]
			Semantic	Syntactic	Total	
NNLM	100	6B	34.2	64.5	50.8	14 x 180
CBOW	1000	6B	57.3	68.9	63.7	2 x 140
Skip-gram	1000	6B	66.1	65.1	65.6	2.5 x 125

rali

Meta-Embeddings

- **idée** : combiner plusieurs représentations vectorielles pour en créer de nouvelles plus efficaces.
- 2 approches simples mais néanmoins utiles (meilleurs résultats que les représentations isolées) :
  - concaténer les représentations ***[Bollegala and Bao, 2018]***
  - les moyenner (normaliser, padder les représentations de plus faible dimension avec des 0) ***[Coates and Bollegala, 2018]***

rali

Don't count, predict ! *[Baroni et al., 2014]*

- plein de tâches, une étude des méta-paramètres de chaque méthode

name	task	measure	source	soa
rg	relatedness	Pearson	Rubenstein and Goodenough (1965)	Hassan and Mihalcea (2011)
ws	relatedness	Spearman	Finkelstein et al. (2002)	Halawi et al. (2012)
wss	relatedness	Spearman	Agirre et al. (2009)	Agirre et al. (2009)
wsr	relatedness	Spearman	Agirre et al. (2009)	Agirre et al. (2009)
men	relatedness	Spearman	Bruni et al. (2013)	Bruni et al. (2013)
toefl	synonyms	accuracy	Landauer and Dumais (1997)	Bullinaria and Levy (2012)
ap	categorization	purity	Almuhareb (2006)	Rothenhäusler and Schütze (2009)
esslli	categorization	purity	Baroni et al. (2008)	Katrenko and Adriaans (2008)
battig	categorization	purity	Baroni et al. (2010)	Baroni and Lenci (2010)
up	sel pref	Spearman	Padó (2007)	Herdağdelen and Baroni (2009)
mcrae	sel pref	Spearman	McRae et al. (1998)	Baroni and Lenci (2010)
an	analogy	accuracy	Mikolov et al. (2013a)	Mikolov et al. (2013c)
ansyn	analogy	accuracy	Mikolov et al. (2013a)	Mikolov et al. (2013a)
ansem	analogy	accuracy	Mikolov et al. (2013a)	Mikolov et al. (2013c)

Table 1: Benchmarks used in experiments, with type of task, figure of merit (measure), original reference (source) and reference to current state-of-the-art system (soa).

rali

Don't count, predict ! *[Baroni et al., 2014]*

- `cnt` = count vector, `pre` = word2Vec, `dm` = *[Baroni and Lenci, 2010]*, `cw` = *[Collobert et al., 2011]*

	rg	ws	wss	wsr	men	toefl	ap	esslli	battig	up	mcrae	an	ansyn	ansem
best setup on each task														
cnt	74	62	70	59	72	76	66	84	98	41	27	49	43	60
pre	84	75	**80**	**70**	**80**	91	75	86	**99**	41	28	**68**	**71**	**66**
best setup across tasks														
cnt	70	62	70	57	72	76	64	84	98	37	27	43	41	44
pre	83	73	78	68	**80**	86	71	77	98	41	26	67	69	64
worst setup across tasks														
cnt	11	16	23	4	21	49	24	43	38	-6	-10	1	0	1
pre	74	60	73	48	68	71	65	82	88	33	20	27	40	10
best setup on rg														
cnt	(74)	59	66	52	71	64	64	84	98	37	20	35	42	26
pre	(84)	71	76	64	79	85	72	84	98	39	25	66	70	61
other models														
soa	**86**	**81**	77	62	76	**100**	**79**	**91**	96	**60**	**32**	61	64	61
dm	82	35	60	13	42	77	76	84	94	51	29	NA	NA	NA
cw	48	48	61	38	57	56	58	61	70	28	15	11	12	9

Table 2: Performance of count (cnt), predict (pre), dm and cw models on all tasks. See Section 3 and Table 1 for figures of merit and state-of-the-art results (soa). Since dm has very low coverage of the an* data sets, we do not report its performance there.

rali

Don't count, predict ! *[Baroni et al., 2014]*

. . .

we set out to conduct this study because we were annoyed by the triumphalist overtones often surrounding predict models, despite the almost complete lack of proper comparison to count vectors. Our secret wish was to discover that it is all hype, and count vectors are far superior to their predictive counterparts.

. . .

we found that the predict models are so good that , while the triumphalist overtones still sound excessive, there are very good reasons to switch to the new architecture.

rali

Représentation vectorielle binaire (non distributionnelle) *[Faruqui and Dyer, 2015]*

- en utilisant des ressources linguistiques (WordNet, PTB, FrameNet, etc.)
- vecteurs très creux
- comparables en performance aux modèles distributionnels état de l'art entraînés sur des billions de mots
- vecteurs disponibles (pour l'anglais) : `https://github.com/mfaruqui/non-distributional`

rali

Représentation vectorielle binaire (non distributionnelle) *[Faruqui and Dyer, 2015]*

Lexicon	Vocabulary	Features
WordNet	10,794	92,117
Supersense	71,836	54
FrameNet	9,462	4,221
Emotion	6,468	10
Connotation	76,134	12
Color	14,182	12
Part of Speech	35,606	20
Syn. & Ant.	35,693	75,972
Union	119,257	172,418

Table 1: Sizes of vocabualry and features induced from different linguistic resources.

rali

Représentation vectorielle binaire (non distributionnelle) *[Faruqui and Dyer, 2015]*

Noun

- S: (n) movie, **film**, picture, moving picture, moving-picture show, motion picture, motion-picture show, picture show, pic, flick
  - ***direct hyponym*** / *full hyponym*
    - S: (n) telefilm
    - S: (n) feature, feature film
    - S: (n) final cut
    - S: (n) home movie
    - S: (n) collage film
    - S: (n) coming attraction
    - S: (n) shoot-'em-up
    - S: (n) short subject
    - S: (n) documentary, docudrama, documentary film, infotainment
    - S: (n) cinema verite
    - S: (n) film noir
    - S: (n) skin flick
    - S: (n) rough cut
    - S: (n) silent movie, silent picture, silents
    - S: (n) slow motion
    - S: (n) talking picture, talkie
    - S: (n) three-D, 3-D, 3D
    - S: (n) musical, musical comedy, musical theater
  - *part meronym*
  - *domain term category*
  - *direct hypernym* / *inherited hypernym* / *sister term*
  - *derivationally related form*
- S: (n) **film**, cinema, celluloid
- S: (n) **film**, photographic film
- S: (n) **film**
- S: (n) **film**, plastic film

Verb

- S: (v) **film**, shoot, take
- S: (v) **film**

features (binaires) induits pour *film* :

SYNSET.FILM.V.01
SYNSET.FILM.N.01
HYPO.COLLAGEFILM.N.01
HYPER :SHEET.N.06
⋮

rali

Représentation vectorielle binaire (non distributionnelle) *[Faruqui and Dyer, 2015]*

supersenses pour les noms, les verbes et les adjectifs
ex : *lioness* ⇒ SS.NOUN.ANIMAL

color lexique mot-couleur élaboré par crowdsourcing *[Mohammad, 2011]*
ex : *blood* ⇒ COLOR.RED

emotion lexique associant un mot à sa polarité (positif,négatif) et aux émotions (joie, peur, tristesse, etc.), élaboré par crowdsourcing *[Mohammad and Turney, 2013]*
ex : *cannibal* ⇒ POL.NEG, EMO.DISGUST et EMO.FEARCOLOR.RED

pos PTB part-of-speech tags
ex : *love* ⇒ PTB.NOUN, PTB.VERB

rali

Représentation vectorielle binaire (non distributionnelle) *[Faruqui and Dyer, 2015]*

Word	POL.POS	COLOR.PINK	SS.NOUN.FEELING	PTB.VERB	ANTO.FAIR	···	CON.NO
love	1	1	1	1	0		1
hate	0	0	1	1	0		0
ugly	0	0	0	0	1		0
beauty	1	1	0	0	0		1
refundable	0	0	0	0	0		1

Table 2: Some linguistic word vectors. 1 indicates presence and 0 indicates absence of a li feature.

- **note :** difficile à faire pour toutes les langues

Représentation vectorielle binaire (non distributionnelle) *[Faruqui and Dyer, 2015]*

Vector	Length (D)	Params.	Corpus Size	WS-353	RG-65	SimLex	Senti	NP
Skip-Gram	300	$D \times N$	300 billion	65.6	72.8	43.6	**81.5**	80.1
Glove	300	$D \times N$	6 billion	60.5	76.6	36.9	77.7	77.9
LSA	300	$D \times N$	1 billion	**67.3**	77.0	49.6	81.1	79.7
Ling Sparse	172,418	–	–	44.6	**77.8**	56.6	79.4	**83.3**
Ling Dense	300	$D \times N$	–	45.4	67.0	**57.8**	75.4	76.2
Skip-Gram $\oplus$ Ling Sparse	172,718	–	–	67.1	80.5	55.5	82.4	82.8

Table 3: Performance of different type of word vectors on evaluation tasks reported by Spearman's correlation (first 3 columns) and Accuracy (last 2 columns). Bold shows the best performance for a task.

- Skip-Gram : pré-entraîné sur 300B de mots *[Mikolov et al., 2013a]*
- Glove : pré-entraîné sur 6B de mots *[Pennington et al., 2014]*
- LSA : obtenue à partir d'une matrice de co-occurrence calculée sur 1B de mots de Wikipedia *[Turney and Pantel, 2010]*
- Ling Dense : réduction de dimensionnalité avec SVD
- taches : similarité, sent. analysis (positif/négatif), NP-bracketing (*local (phone company)* versus *(local phone) company*)

Retrofitting de vecteurs à une ressource lexico-sémantique *[Faruqui et al., 2015a]*

- étape de post-traitement applicable à n'importe quelle représentation vectorielle de mots
- rapide (5 secondes pour 100k mots et dimension 300)
- **idée :** utiliser les informations lexico-sémantiques d'une ressource pour améliorer une représentation existante
- **comment :** encourager que les mots de distance similaire dans la représentation apprise soit proche de la représentation induite de la ressource (encodée sous forme de graphe).

rali

Une communauté qui s'organise

- ***[Faruqui and Dyer, 2014]***
  - des embeddings déjà entraînés, une suite de tests qui peuvent s'exécuter (similarité, analogie, complétion, etc.), une interface de visualisation
  - un lien qui n'est plus valide : `http://wordvectors.org/demo.php`
- Glue ***[Wang et al., 2018]***
  - 9 taches de compréhension (classification de phrases ou paires de phrases)
  - observer les gains en performance depuis le lancement !
  - voir aussi superGlue !

rali

Mikolov strikes again *[Mikolov et al., 2013b]*

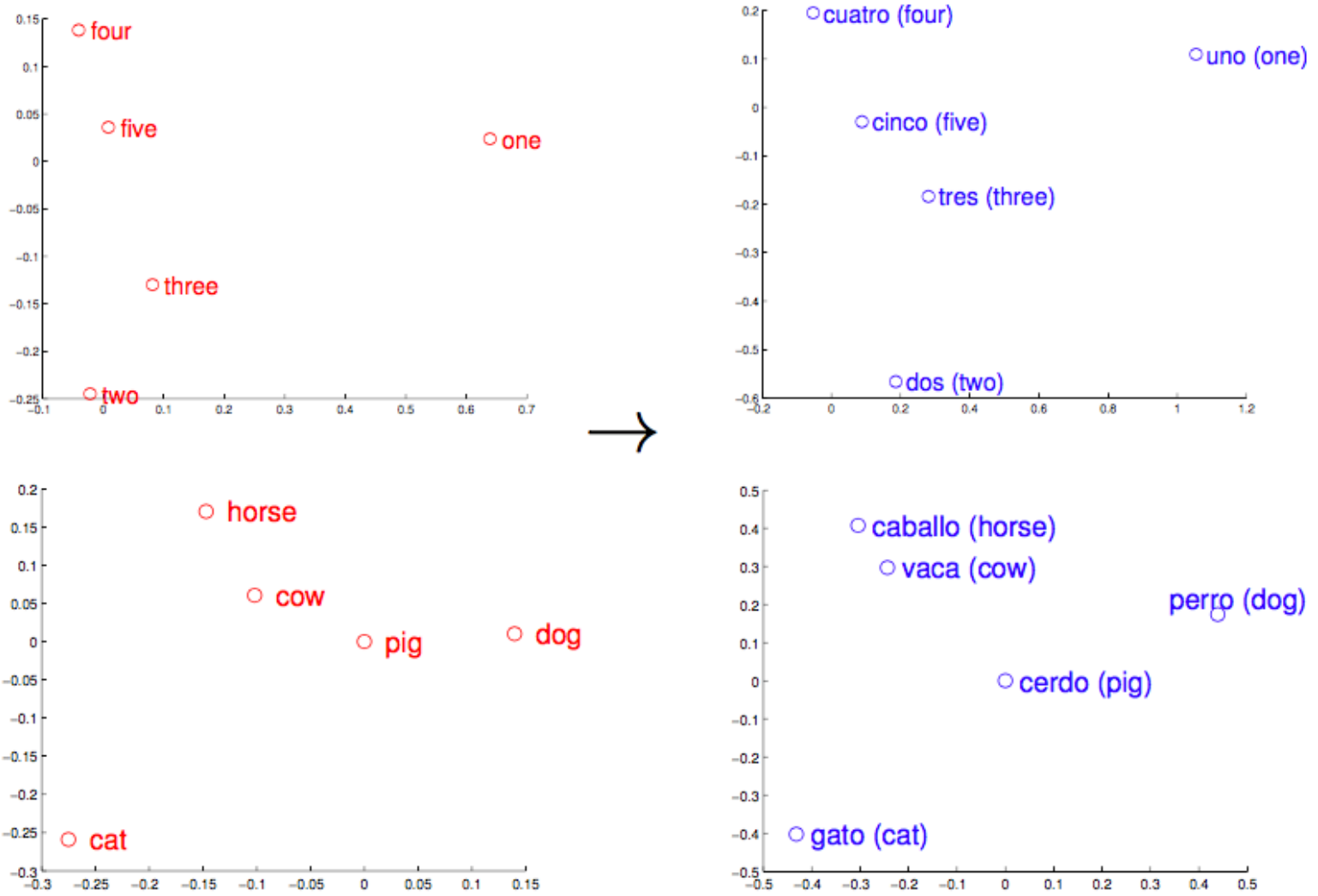


Figure 1: Distributed word vector representations of numbers and animals in English (left) and Spanish (right). The five vectors in each language were projected down to two dimensions using PCA, and then manually rotated to accentuate

Mikolov strikes again *[Mikolov et al., 2013b]*

- on peut apprendre une transformation linéaire (rotation + scaling) d'un espace vers un autre avec un lexique bilingue $\{(x_i, z_i)\}$:

$$\hat{W} = \min_W \Sigma_i \|W\overline{x}_i - \overline{z}_i\|^2$$

  où $\overline{x}_i$ et $\overline{z}_i$ désignent respectivement la représentation vectorielle source de x_i et cible de z_i

- W optimisée par descente de gradient sur un lexique d'environ 5k paires de mots
- au moment du test : *traduire* un mot x par $\hat{z}$:

$$\hat{z} = \operatorname*{argmax}_z \cos(\overline{z}, \hat{W}\overline{x})$$

rali

Mikolov strikes again *[Mikolov et al., 2013b]*

Table 1: The sizes of the monolingual training datasets from WMT11. The vocabularies consist of the words that occurred at least five times in the corpus.

Language	Training tokens	Vocabulary size
English	575M	127K
Spanish	84M	107K
Czech	155M	505K

- 6K des most sources les plus fréquents traduits par GoogleTrans
  - premières 5K entrées pour calculer $\hat{W}$
  - 1K suivantes pour les tests
- baselines : edit-distance, $\epsilon-$Rapp

Mikolov strikes again *[Mikolov et al., 2013b]*

Table 2: Accuracy of the word translation methods using the WMT11 datasets. The Edit Distance uses morphological structure of words to find the translation. The Word Co-occurrence technique based on counts uses similarity of contexts in which words appear, which is related to our proposed technique that uses continuous representations of words and a Translation Matrix between two languages.

Translation	**Edit Distance**		**Word Co-occurrence**		**Translation Matrix**		**ED + TM**		**Coverage**
	P@1	P@5	P@1	P@5	P@1	P@5	P@1	P@5	
En → Sp	13%	24%	19%	30%	33%	51%	43%	60%	92.9%
Sp → En	18%	27%	20%	30%	35%	52%	44%	62%	92.9%
En → Cz	5%	9%	9%	17%	27%	47%	29%	50%	90.5%
Cz → En	7%	11%	11%	20%	23%	42%	25%	45%	90.5%

rali

Plus de données ? (Google News)

- même split : 5K train / 1K test

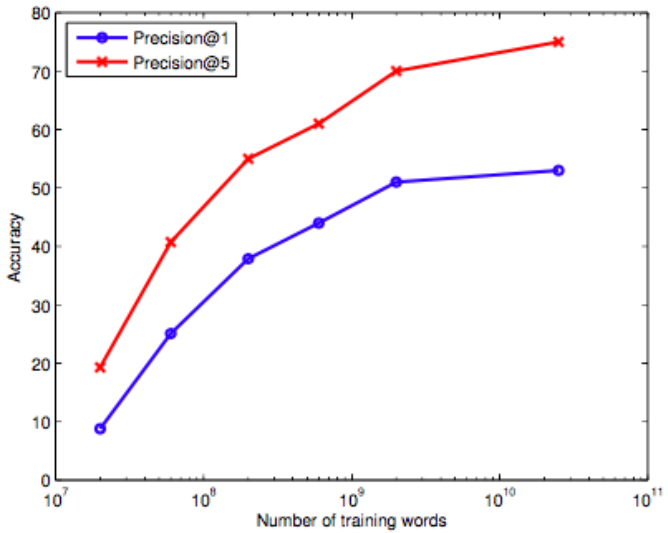


Figure 3: The Precision at 1 and 5 as the size of the monolingual training sets increase (EN→ES).

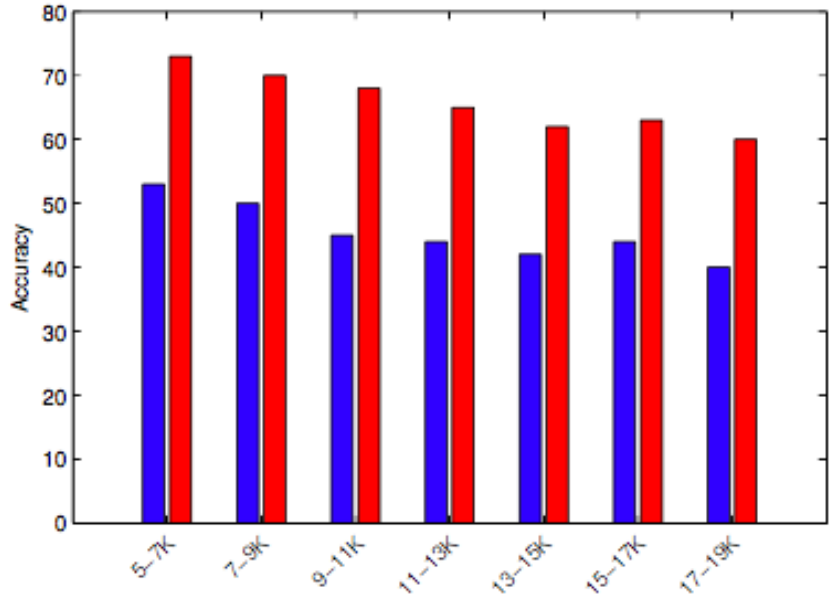


Figure 4: Accuracies of translation as the word frequency decreases. Here, we measure the accuracy of the translation on disjoint sets of 2000 words sorted by frequency, starting from rank 5K and continuing to 19K. In all cases, the linear transformation was trained on the 5K most frequent words and their translations. EN→ES.

rali

Plan

Approche distributionnelle

Before Deep : modèle vectoriel

And then came the “Deep”
- Word2Vec
- Analogie
- Meta-embeddings
- Évaluation
- Idées intéressantes
- Le cas bilingue

Évaluation

Plongements contextuels

rali

Sur la difficulté d'évaluer sans biais

[Levy et al., 2015]

- comparent 4 approches : matrice de co-occurrence (PMI), SVD, Skip-Gram, et GloVe
- étudient leurs paramètres en détail
- adaptent des choix faits dans Skip-Gram à d'autres méthodes lorsque possible
- **Bilan :**
  - match nul en performance (pas d'avantage clair d'une approche sur une autre)
  - Skip-Gram se comporte mieux (temps/mémoire) que les autres approches

rali

Sur la difficulté d'évaluer sans biais

[Levy et al., 2015]

Method	WordSim Similarity	WordSim Relatedness	Bruni et al. MEN	Radinsky et al. M. Turk	Luong et al. Rare Words	Hill et al. SimLex	Google Add / Mul	MSR Add / Mul
PPMI	.755	**.688**	.745	**.686**	.423	.354	.553 / **.629**	.289 / .413
SVD	**.784**	.672	**.777**	.625	**.514**	.402	.547 / .587	.402 / .457
SGNS	.773	.623	.723	.676	.431	**.423**	.599 / .625	.514 / .546
GloVe	.667	.506	.685	.599	.372	.389	.539 / .563	.503 / **.559**
CBOW	.766	.613	.757	.663	.480	.412	.547 / .591	.557 / **.598**

Table 3: Performance of each method across different tasks using `word2vec`'s recommended configuration: `win` = 2; `dyn` = with; `sub` = dirty; `neg` = 5; `cds` = 0.75; `w+c` = only w; `eig` = 0.0. CBOW is presented for comparison.

Method	WordSim Similarity	WordSim Relatedness	Bruni et al. MEN	Radinsky et al. M. Turk	Luong et al. Rare Words	Hill et al. SimLex	Google Add / Mul	MSR Add / Mul
PPMI	.755	**.697**	.745	.686	.462	.393	.553 / .679	.306 / .535
SVD	**.793**	.691	**.778**	.666	**.514**	.432	.554 / .591	.408 / .468
SGNS	**.793**	.685	.774	**.693**	.470	**.438**	.676 / **.688**	.618 / **.645**
GloVe	.725	.604	.729	.632	.403	.398	.569 / .596	.533 / .580

Table 4: Performance of each method across different tasks using the best configuration for that method and task combination, assuming `win` = 2.

rali

Exemple d'observation *[Levy et al., 2015]*

- dans l'approche matrice de co-occurences, un mot w et son contexte c est noté :

$$PMI(w,c) = \log \frac{p(w,c)}{p(w)p(c)}$$

  - une approche courante est de mettre à 0 les valeurs de PMI lorsque $\#(w,c) = 0$ (plutôt que $-\infty$)
  - une autre est de prendre : $PPMI(w,c) = max(PMI(w,c), 0)$

- adaptation de choix faits dans Skip-Gram :

  - $$SPPMI(w,c) = \max(PMI(w,c) - logk, 0)$$
  - sampling des k examples négatifs (lissés avec $\alpha = 0.75$)

$$PMI_\alpha(w,c) = \log \frac{P(w,c)}{p(w).P_\alpha(c)} \text{ avec } P_\alpha(c) = \frac{\#(c)^\alpha}{\sum_c \#(c)^\alpha}$$

[Schnabel et al., 2015]

- recommandent de ne pas utiliser une tâche extrinsèque pour évaluer des embeddings pré-entraînés

	dev	test	p-value
Baseline	94.18	93.78	0.000
Rand. Proj.	94.33	93.90	0.006
GloVe	94.28	93.93	0.015
H-PCA	94.48	93.96	0.029
C&W	**94.53**	**94.12**	
CBOW	94.32	93.93	0.012
TSCCA	**94.53**	94.09	0.357

Table 4: F1 chunking results using different word embeddings as features. The p-values are with respect to the best performing method.

	test	p-value
BOW (baseline)	88.90	$7.45 \cdot 10^{-14}$
Rand. Proj.	62.95	$7.47 \cdot 10^{-12}$
GloVe	74.87	$5.00 \cdot 10^{-2}$
H-PCA	69.45	$6.06 \cdot 10^{-11}$
C&W	72.37	$1.29 \cdot 10^{-7}$
CBOW	**75.78**	
TSCCA	75.02	$7.28 \cdot 10^{-4}$

Table 5: F1 sentiment analysis results using different word embeddings as features. The p-values are with respect to the best performing embedding.

rali

[Antoniak and Mimno, 2018]

- word2vec skipgram relancé plusieurs fois avec les mêmes paramètres

Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7
viability	fetus	trimester	surgery	trimester	pregnancies	abdomen
pregnancies	pregnancies	surgery	visit	surgery	occupation	tenure
abortion	gestation	visit	therapy	incarceration	viability	stepfather
abortions	kindergarten	tenure	pain	visit	abortion	wife
fetus	viability	workday	hospitalization	arrival	tenure	groin
gestation	headaches	abortions	neck	pain	visit	throat
surgery	pregnant	hernia	headaches	headaches	abortions	grandmother
expiration	abortion	summer	trimester	birthday	pregnant	daughter
sudden	pain	suicide	experiencing	neck	birthday	panic
fetal	bladder	abortion	medications	tenure	fetus	jaw

Table 5: The 10 closest words to the query term `pregnancy` are highly variable. None of the words shown appear in every run. Results are shown across runs of the BOOTSTRAP setting for the full corpus of the *9th Circuit*, the whole document size, and the SGNS model.

Et pour les mots peu fréquents ?

[Jakubina and Langlais, 2017]

donut		beigne		
context	-	aromatisé (0.05)	donut (0.05)	beignet (0.04)
embedding	-	liper (0.54)	babalous (0.53)	savonnettes (0.52)
brilliantly		brillamment		
context	-	imaginatif (0.05)	captivant (0.05)	rusé (0.05)
embedding	-	éclatant (0.69)	pathétique (0.67)	émouvant (0.66)
gentle		doucet,	doux,	délicat,
context	-	enjoué (0.05)	serviable (0.05)	affable (0.04)
embedding	-	colérique (0.76)	enjoué (0.75)	espiègle (0.75)
pathologically		pathologiquement		
context	-	cordonale (0.05)	pathologique (0.05)	diagnostiqué (0.05)
embedding	-	psychosexuel (0.60)	psychoaffectif (0.60)	piloérection (0.59)

Fig. 1. Top-3 candidates produced by the two best approaches for a few test words.

rali

Et pour les mots peu fréquents ?

	1k-low			1k-high		
	TOP@1	TOP@5	TOP@20	TOP@1	TOP@5	TOP@20
`embedding`	**2,2**	**6,1**	**11,9**	**21,7**	**34,2**	**44,9**
`context`	2,0	4,3	7,6	19,0	32,7	44,3
`document`	0,7	2,3	5,0	—	—	—
oracle	4,6	—	19,0	31,8	—	57,6

- Wikipedia dump de juin 2013 (EN : 3.5M, FR : 1.3M articles)
- V_{EN} = 7.3M, V_{FR} = 3.6M
- 2 test sets : 1k-low (1k mots rares) 1k-high (1k mots non rares)
- rare = freq $<$ 26 (92% des mots de V_{EN})

rali

Plan

rali

todo

ELMO, BERT (et ses variantes), XLM, XLN

Al-Rfou, R., Perozzi, B., and Skiena, S. (2013).
Polyglot : Distributed word representations for multilingual nlp.
In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria. Association for Computational Linguistics.

Antoniak, M. and Mimno, D. (2018).
Evaluating the stability of embedding-based word similarities.
Transactions of the Association for Computational Linguistics, 6 :107–119.

Baroni, M., Dinu, G., and Kruszewski, G. (2014).
Don't count, predict ! a systematic comparison of context-counting vs. context-predicting semantic vectors.
In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 238–247, Baltimore, Maryland. Association for Computational Linguistics.

Baroni, M. and Lenci, A. (2010).

rali

Distributional memory : A general framework for corpus-based semantics.
Comput. Linguist., 36(4) :673–721.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016).
Enriching word vectors with subword information.
arXiv preprint arXiv :1607.04606.

Bollegala, D. and Bao, C. (2018).
Learning word meta-embeddings by autoencoding.
In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1650–1661. Association for Computational Linguistics.

Chandar, A. P. S., Lauly, S., Larochelle, H., Khapra, M. M., Ravindran, B., Raykar, V. C., and Saha, A. (2014).
An autoencoder approach to learning bilingual word representations.
CoRR.

Coates, J. and Bollegala, D. (2018).

rali

Frustratingly easy meta-embedding – computing meta-embeddings by averaging source word embeddings.
In *Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 2 (Short Papers)*, pages 194–198.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011).
Natural language processing (almost) from scratch.
Journal of Machine Learning Research, 12 :2493–2537.

Coulmance, J., Marty, J., Wenzek, G., and Benhalloum, A. (2016).
Trans-gram, fast cross-lingual word-embeddings.
CoRR, abs/1601.02502.

Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., and Smith, N. A. (2015a).
Retrofitting word vectors to semantic lexicons.
In *Proceedings of NAACL*.

Faruqui, M. and Dyer, C. (2014).

rali

Community evaluation and exchange of word vectors at wordvectors.org.
In *Proceedings of ACL : System Demonstrations*.

Faruqui, M. and Dyer, C. (2015).
Non-distributional word vector representations.
In *Proceedings of ACL*.

Faruqui, M., Tsvetkov, Y., Yogatama, D., Dyer, C., and Smith, N. A. (2015b).
Sparse overcomplete word vector representations.
In *Proceedings of ACL*.

Golub, G. H. and Van Loan, C. F. (1996).
Matrix Computations (3rd Ed.).
Johns Hopkins University Press.

Gouws, S., Bengio, Y., and Corrado, G. (2015).
Bilbowa : Fast bilingual distributed representations without word alignments.
In *ICML*.

rali

Jakubina, L. and Langlais, P. (2017).
Reranking translation candidates produced by several bilingual word similarity sources.
In *15th Conference of the European Chapter of the Association for Computational Linguitics*, volume 2, Short Papers, pages 605–611.

Jurafsky, D. and Martin, J. H. (2015).
Speech and language processing.
(3rd ed. draft).

Lee, D. D. and Seung, H. S. (1999).
Learning the parts of objects by non-negative matrix factorization.
Nature, 401(6755) :788–791.

Levy, O. and Goldberg, Y. (2014).
Neural word embedding as implicit matrix factorization.
In *Advances in Neural Information Processing Systems 27*, pages 2177–2185.

rali

Levy, O., Goldberg, Y., and Dagan, I. (2015).
Improving distributional similarity with lessons learned from word embeddings.
Transactions of the Association for Computational Linguistics, 3 :211–225.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a).

Efficient estimation of word representations in vector space.
CoRR, abs/1301.3781.

Mikolov, T., Le, Q. V., and Sutskever, I. (2013b).
Exploiting similarities among languages for machine translation.
CoRR, abs/1309.4168.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013c).
Distributed representations of words and phrases and their compositionality.
CoRR, abs/1310.4546.

Mikolov, T., tau Yih, W., and Zweig, G. (2013d).
Linguistic regularities in continuous space word representations.
In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL-HLT-2013)*.

Mohammad, S. (2011).
Colourful language : Measuring word-colour associations.
In *2Nd Workshop on Cognitive Modeling and Computational Linguistics*, CMCL '11, pages 97–106.

Mohammad, S. and Turney, P. D. (2013).
Crowdsourcing a word-emotion association lexicon.
CoRR.

Pennington, J., Socher, R., and Manning, C. D. (2014).
Glove : Global vectors for word representation.
In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Salton, G. (1975).

rali

Dynamic information and library processing / Gerard Salton.
Prentice-Hall Englewood Cliffs, N.J.

Schnabel, T., Labutov, I., Mimno, D. M., and Joachims, T. (2015).
Evaluation methods for unsupervised word embeddings.
In Màrquez, L., Callison-Burch, C., Su, J., Pighin, D., and Marton, Y., editors, *EMNLP*, pages 298–307. The Association for Computational Linguistics.

Schwenk, H., Kiela, D., and Douze, M. (2019).

Turney, P. D. (2005).
Measuring semantic similarity by latent relational analysis.
CoRR.

Turney, P. D. and Pantel, P. (2010).
From frequency to meaning : Vector space models of semantics.

J. Artif. Int. Res., 37(1) :141–188.

rali

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018).
GLUE : A multi-task benchmark and analysis platform for natural language understanding.
In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP : Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.