Word embeddings evaluation and combination

Sahar GHANNAY[1], Benoit Favre[2], Yannick Estève[1] and Nathalie Camelin[1]

[1]LIUM-University of Le Mans

[2]Aix-Marseille Université, CNRS, LIF UMR 7279, 13000, Marseille, France

Introduction

Word Embeddings:

- Successfully used in several Natural Language Processing (NLP) and speech processing tasks
- Different approaches are introduced to calculate them through neural networks
- Their evaluation needs to be more studied

Goal:

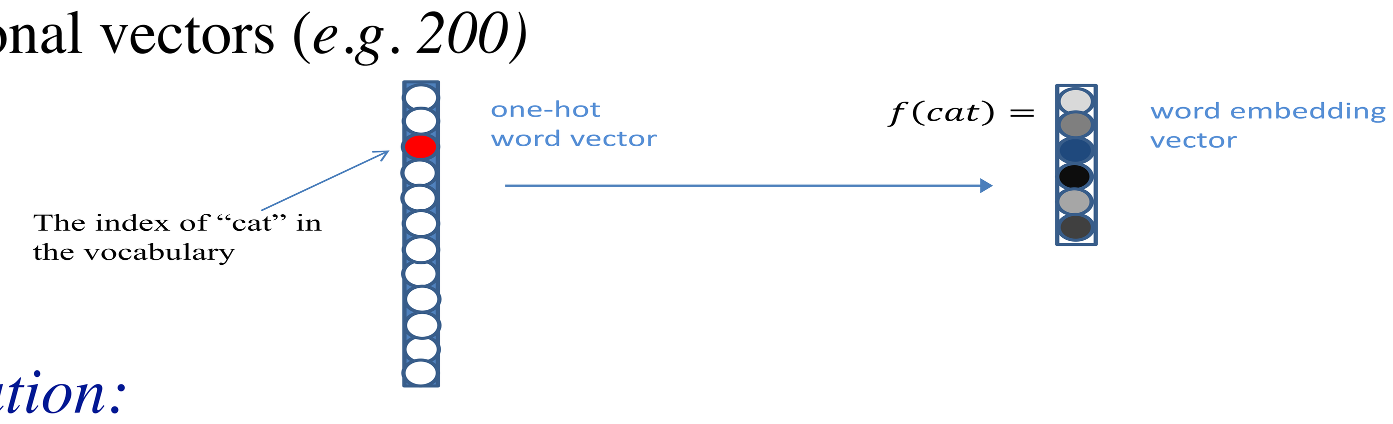
- Rigorous comparison of the performances of different kinds of word embeddings on **NLP**, **analogical** and **similarity** tasks.
- Word embeddings combination

→ Looking for the most effective word embeddings!

Word Embeddings

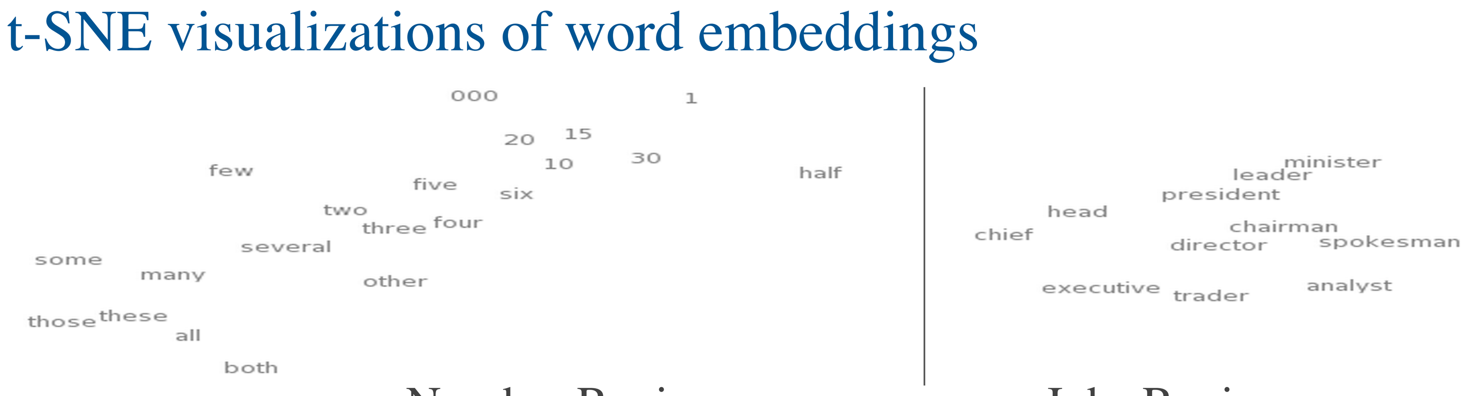
Definition:

f: words→ $\mathbb{R}^n$ is a function for mapping words to low-dimensional vectors (*e.g. 200*)



Visualization:

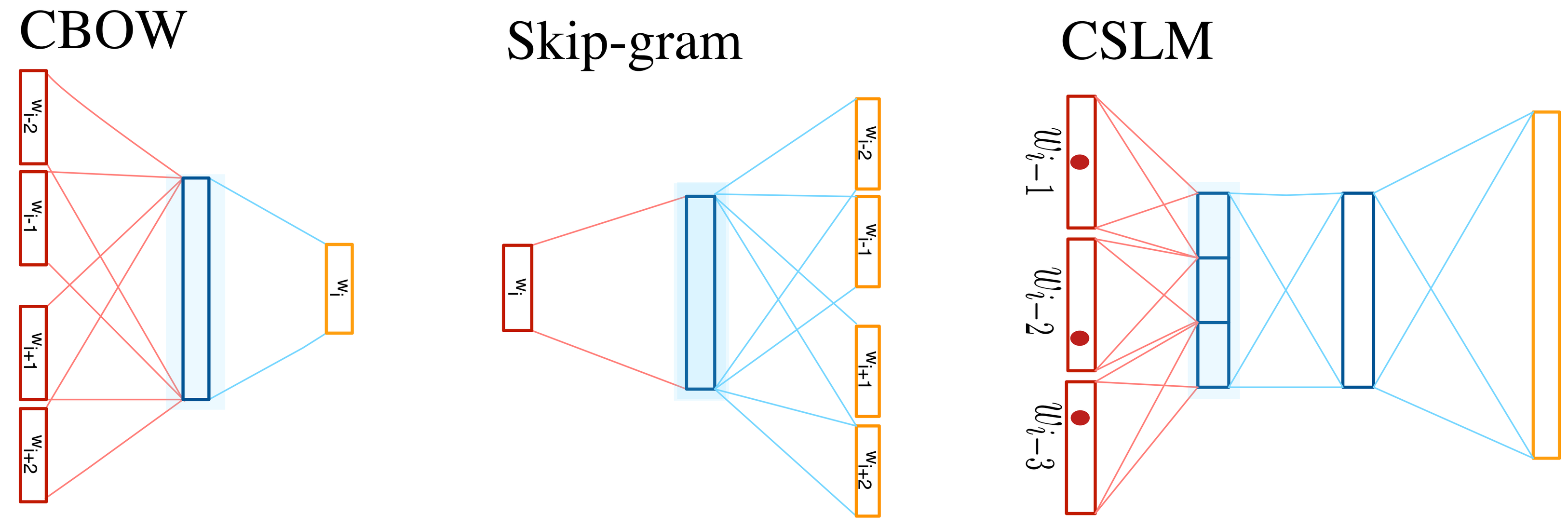
t-SNE visualizations of word embeddings



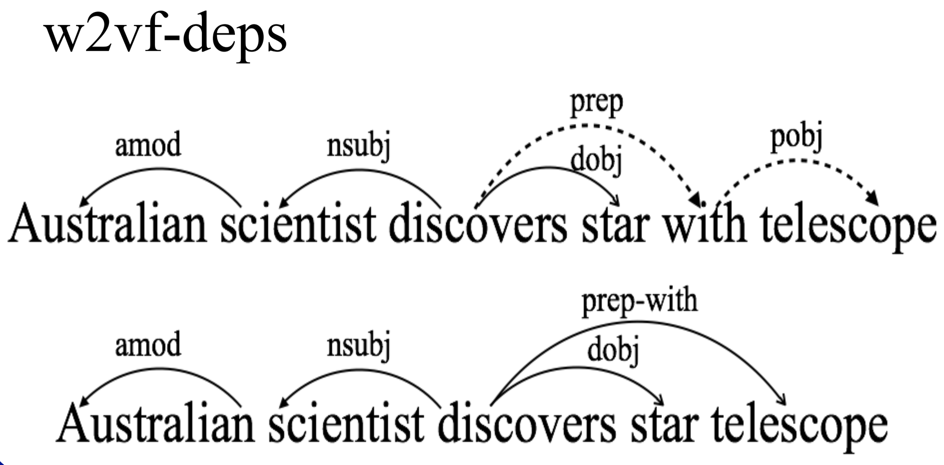
Number Region | Jobs Region

Approaches:

CBOW | Skip-gram | CSLM



w2vf-deps



GloVe

1. Co-occurence matrix X from the training set
2. Factorize X to get vectors

Benchmark tasks

1. NLP Tasks:

- Part-of-speech tagging (POS): syntactic roles (noun, adverb, *etc.*)
- Chunking (CHK): syntactic constituent (noun phrase, verb phrase, *etc.*)
- Named Entity recognition (NER): person, company, *etc.*
- Mention detection (MENT): begin, inside, and outside

2. Similarity task

Measuring:

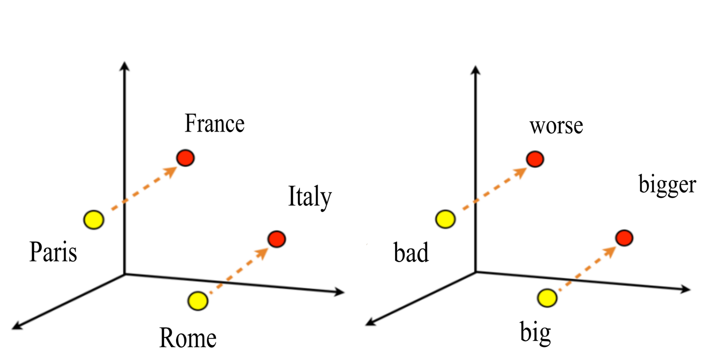
Similarity	How similar is pizza to pasta?
Relatedness	How related is pizza to Italy?

3. Analogical task:

Answering questions:

Semantic: Paris:France → Rome:?

Syntactic: bad:worse → big:?



Experiments

1. Setup:

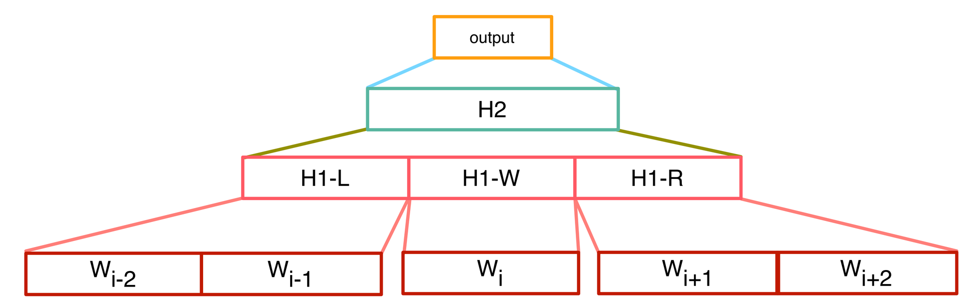
Word embeddings:

Data: Gigaword corpus composed of 4 billion words

Vocabulary size: 239K words

NLP tasks:

Neural architecture:



Analogical task:

Evaluation sets:

- 8,869 semantic
- 10,675 syntactic

Similarity task:

Evaluation sets:

- WordSim353
- RW
- MEN

2. Results:

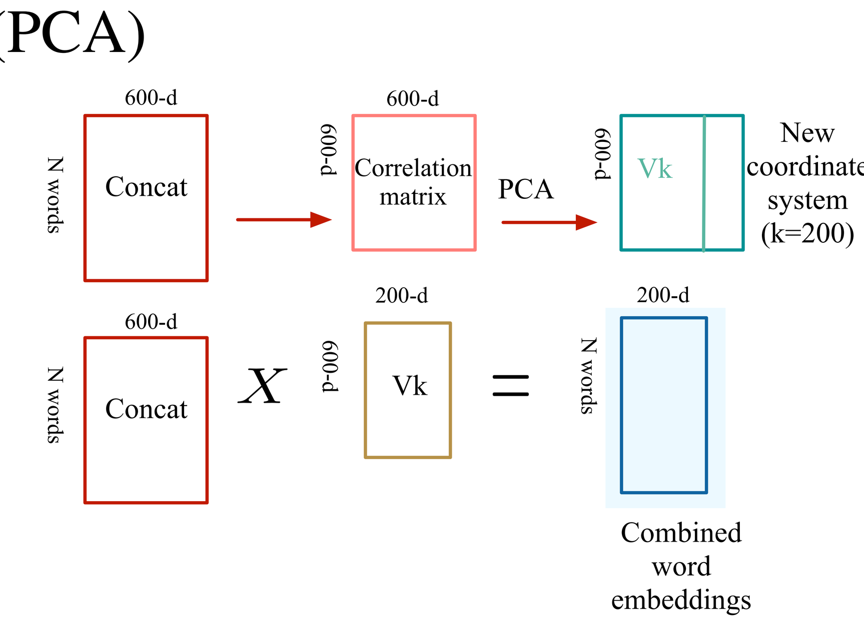
Embeddings	NLP Tasks POS	CHK	NER	MENT	Similarity Task WS353	RW	MEN	Analogical Task
	Acc.	F1			Spearman's ρ			Acc.
CBOW	96.01	90.48	78.32	55.49	**59.0**	46.5	60.9	57.2
Skip-gram	96.43	89.64	77.65	57.80	55.8	**50.2**	**66.2**	62.3
GloVe	95.79	86.90	76.45	54.49	53.3	41.0	66.0	**65.5**
CSLM	96.24	90.11	76.20	57.34	47.8	43.4	48.2	27.4
w2vf-deps	**96.66**	**92.02**	**79.37**	**58.06**	52.3	43.5	55.7	42.70

Combined word embeddings

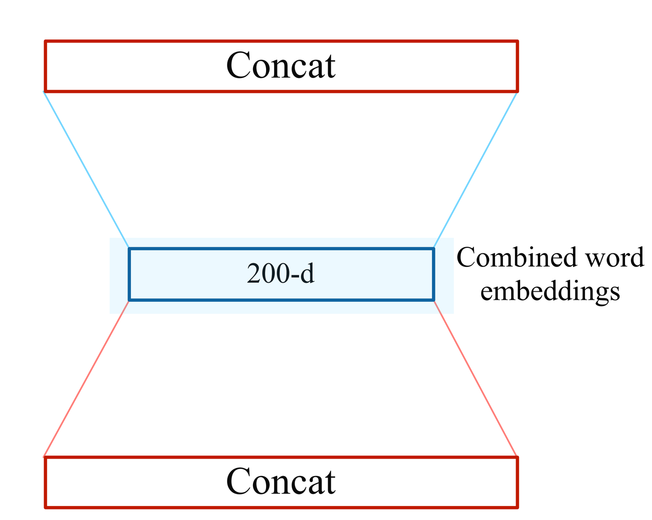
1. Combination approaches:

1) Simple concatenation (Concat)

2) Principal component Analysis (PCA)



3) Auto-encoder



2. Performance of combined word embeddings:

Dim.	Embeddings	NLP Tasks POS	CHK	NER	MENT	Similarity Task WS353	RW	MEN	Analogical Task
		Acc.	F1			Spearman's ρ			Acc.
600	Best-Concat	**96.67**	91.88	**81.06**	**58.20**	**57.0**	48.6	**69.4**	**71.4**
200	Best-PCA	96.45	90.13	**79.66**	**60.22**	**57.9**	49.5	**71.3**	**70.7**
	Best-AutoE	96.64	91.35	**80.43**	**60.39**	55.8	44.6	64.9	62.0

Conclusions

- ✚ w2vf-deps is the best for NLP tasks
- ✚ Skip-gram is the best for similarity task
- ✚ GloVe is the best for analogical task
- ✚ The combination of w2vf-deps, Skip-gram and GloVe yields significant improvements.

→ Best-AutoE achieves best results on NLP tasks

→ Building an effective word embedding can be reached by the combination of the efficient embeddings in each task through PCA or concatenation