

# Multilingual Distributional Semantics

Benno Kruit    Sara Veldhoen

January 13, 2015

# Outline

Introduction - related work

Introduction -  
related work

Multilingual DM

Multilingual DM

Multilingual Dbow

Multilingual Dbow

Evaluation

Evaluation

Results

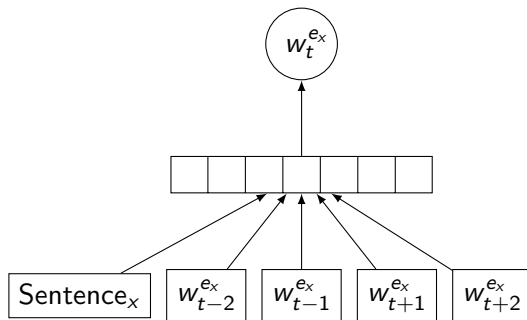
Results

Graphics and concluding words

Graphics and  
concluding words

F1 baseline

# Multilingual DM



**Figure :** Bilingual distributed memory. The same architecture is trained with English context and word prediction replaced by the other language(s).

# Multilingual Dbow

Multilingual  
Distributional  
Semantics

Kruit, Veldhoen

Introduction -  
related work

Multilingual DM

Multilingual Dbow

Evaluation

Results

Graphics and  
concluding words

F1 baseline

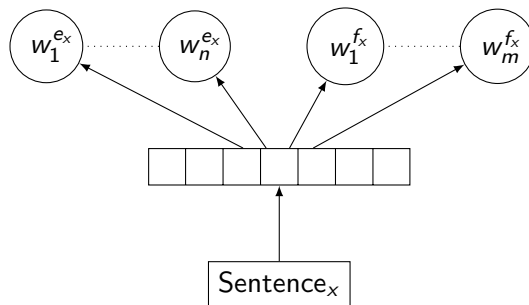


Figure : Bilingual dbow

# Multilingual Dbow

Multilingual  
Distributional  
Semantics

Kruit, Veldhoen

Introduction -  
related work

Multilingual DM

Multilingual Dbow

Evaluation

Results

Graphics and  
concluding words

F1 baseline

- ▶ Training a single embedding for parallel sentences
- ▶ Word embeddings are not trained
- ▶ Can be extended to more than two languages

- ▶ Use the sentence embeddings to obtain word vector:

$$emb(w) = \frac{1}{freq(w, D)} \sum_{s \in D} freq(w, s) emb(s)$$

- ▶ Quite good performance (as we will see later)

# Multilingual Distributional Semantics

## Introduction - related work

## Multilingual Dbow

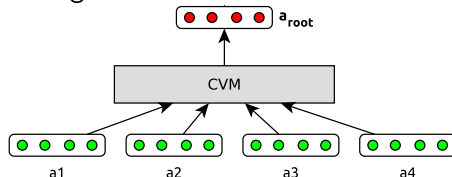
## Results

## Graphics and concluding words

F1 baseline

-

- ▶ We could have a similar training procedure
- ▶ Only: we are not training the sentences, but assume fixed 'gold standard' sentence embeddings



- ▶ So, we could plug in any compositional model



- ▶ Training word embeddings: on Europarl data (50k or 500k sentences)
- ▶ Monolingual (English) evaluation: analogy task
- ▶ Crosslingual evaluation: document classification

## Crosslingual Document classification:

- ▶ Given word embeddings, obtain document representation for train and test documents in all languages

$$emb(doc) = \sum_{w \in doc} idf(w) * emb(w)$$

- ▶ Train a classifier (averaged perceptron) on the training document representations for one language
- ▶ Test classifier performance on the test document representations for another language

RCV (Reuters) data:

- ▶ English-German
- ▶ Multiclass classification:  
each document is assigned a single class (topic)
- ▶ Performance measure: accuracy
- ▶ Baseline: majority class

TED data:

- ▶ Many languages
- ▶ Binary classification: each class (topic) has positive and negative examples
- ▶ Performance measure: F1 score
- ▶ Baseline: ??

## Monolingual evaluation on English:

Setting	Length	RCV (1000) accuracy	TED F1
Baseline		.468	.118
I-Matrix	40	.861	.154
Paragraph mono	256		
Paragraph bi	256		
Paraword mono	256		
Paraword bi	256	.898	.216
Paraword multi	256	.903	.245
Google News	300	.951	.486

- ▶ Word vectors as average of the dbow-trained sentences they occur in.
- ▶ Sentences trained on 50k Europarl data in specified languages.
- ▶ Mono- and bilingual evaluation on TED data (F1 scores):

Sentences trained on:	Classification train-test			
	EN-EN	DE-DE	EN-DE	DE-EN
EN	.186	.134	.084	.153
DE	.132	.091	.076	.132
DE-EN	.216	.189	.201	.220
multi	.404	.368	.387	.339

- ▶ Word vectors as average of the dbow-trained sentences they occur in.
- ▶ Sentences trained on 50k Europarl data in specified languages.
- ▶ Mono- and bilingual evaluation on RCV data with 10000 training examples (accuracy):

Sentences trained on:	Classification train-test			
	EN-EN	DE-DE	EN-DE	DE-EN
EN	.866	.456	.398	.661
DE	.850	.449	.389	.704
DE-EN	.898	.467	.415	.708
multi	.903	.473	.431	.705

# Results

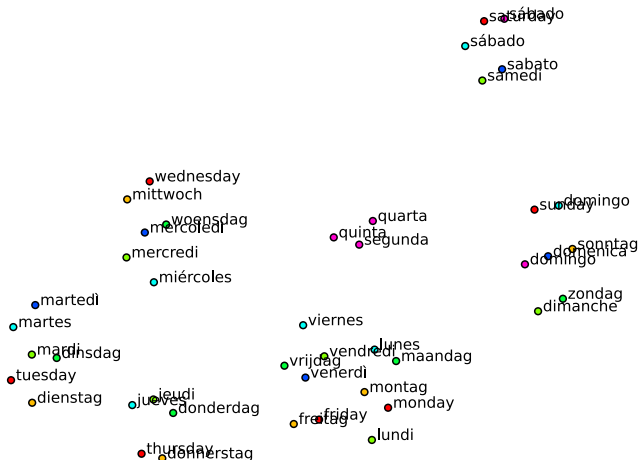
- ▶ Word vectors as average of the dbow-trained sentences they occur in.
- ▶ Sentences trained on 50k Europarl data in all languages.
- ▶ multilingual evaluation on TED data (F1 scores):

F1 Trained on	Tested on						
	de	en	es	fr	it	nl	pb
de	0,36753	0,33879	0,4028	0,368	0,28221	0,37315	0,31928
en	0,38686	0,40439	0,38929	0,32149	0,35167	0,37379	0,35102
es	0,39853	0,30125	0,42759	0,38709	0,3536	0,36173	0,35515
fr	0,39842	0,41654	0,54487	0,40679	0,38499	0,33246	0,40565
it	0,40612	0,40535	0,37698	0,43608	0,37289	0,40004	0,35872
nl	0,4265	0,39681	0,41736	0,39255	0,41243	0,42775	0,32053
pb	0,40317	0,33343	0,36931	0,35449	0,37403	0,40549	0,31451



# Graphics and concluding words

Words from *multilingual* dbow paragraphs (7 languages)







# F1 baseline

$$Prec = \frac{TP}{TP + FP},$$

$$Rec = \frac{TP}{TP + FN},$$

$$Acc = \frac{TP + TN}{TP + FP + TN + FN}$$

Majority class:

$$neg > pos \rightarrow \begin{cases} Acc = \frac{TP+TN}{TP+FP+TN+FN} = \frac{TN}{TN+FN} = \frac{neg}{total} \\ Prec = \frac{TP}{TP+FP} = 0 \rightarrow F1 = 0 \end{cases}$$

# F1 baseline

Now assume a stochastic classifier:

$$P = P(pos) = \frac{pos}{total}, P(neg) = 1 - P$$

$$pos = P * |X|, neg = (1 - P) * |X|$$

$$TP = P * pos = P^2 * |X|$$

$$FP = P * neg = P * (1 - P) * |X|$$

$$FN = (1 - P) * pos = (1 - P) * P * |X|$$

$$\begin{aligned} F1 &= \frac{2 * TP}{2 * TP + FN + FP} \\ &= \frac{2 * P^2 * |X|}{2 * P * P * |X| + (1 - P) * P * |X| + (1 - P) * P * |X|} \\ &= \frac{2 * P^2}{2 * P^2 + (1 - P) * P + (1 - P) * P} \\ &= \frac{2 * P}{2 * P + (1 - P) + (1 - P)} = \frac{2P}{2} = P \end{aligned}$$