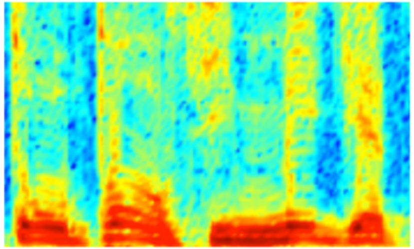
Training Neural Word Embeddings for Transfer Learning and Translation

Stephan Gouws

stephan@ml.sun.ac.za

Introduction

AUDIO



Audio Spectrogram

DENSE

IMAGES

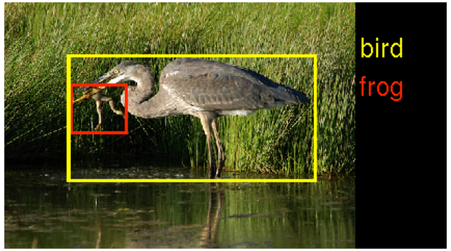


Image pixels

DENSE

TEXT

0	0	0	0.2	0	0.7	0	0	0	...	...

Word, context, or document vectors

SPARSE

Motivation

Monolingual word embeddings

The NLP community has developed good features for several tasks, but finding

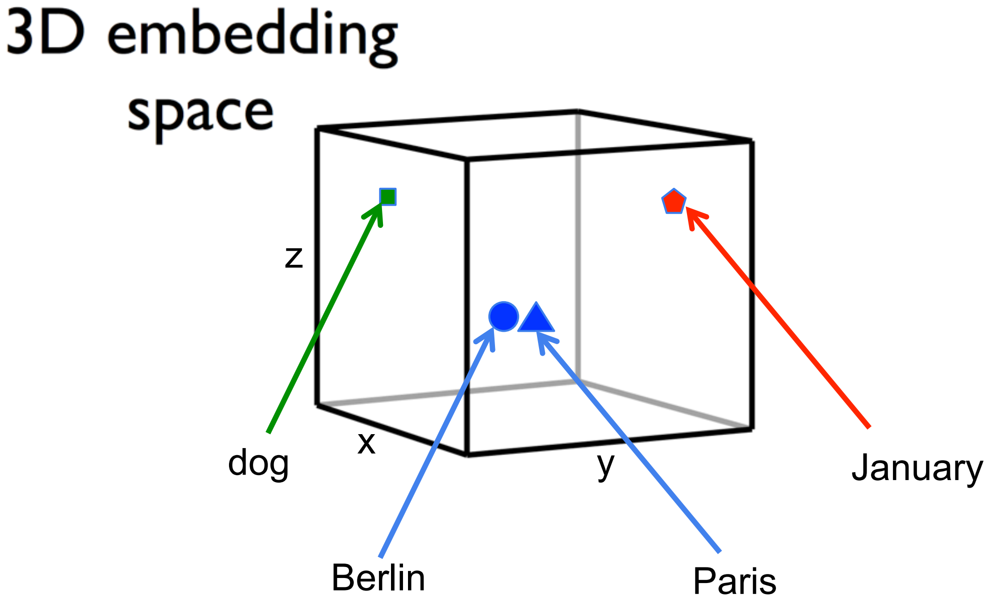
- **task**-invariant (POS tagging, NER, SRL); **AND**
- **language**-invariant (English, Danish, Afrikaans,..)

Cross-lingual word embeddings (this talk)

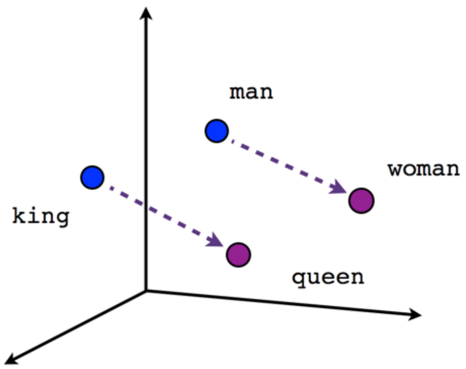
features is non-trivial and time-consuming (been trying for 20+ years).

Learn word-level features which generalize across tasks ***and*** languages.

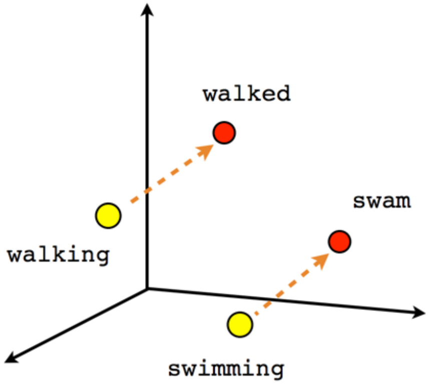
Word Embeddings



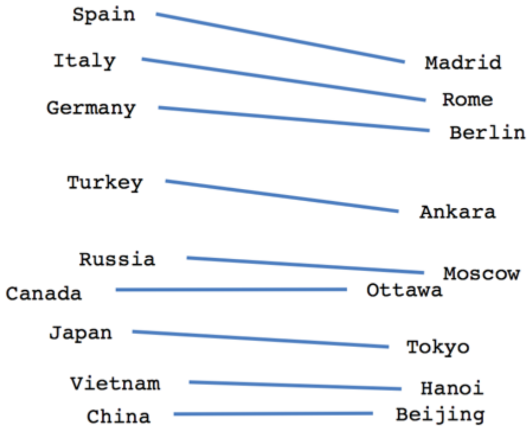
Word Embeddings Capture Interesting, Universal Features



Male-Female



Verb tense

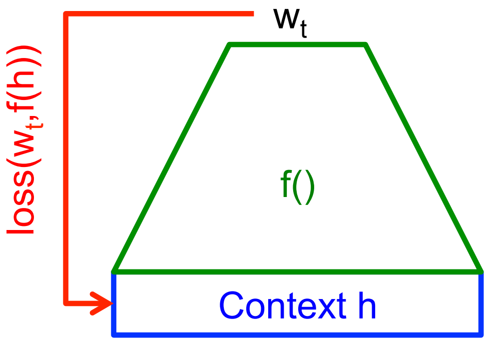


Country-Capital

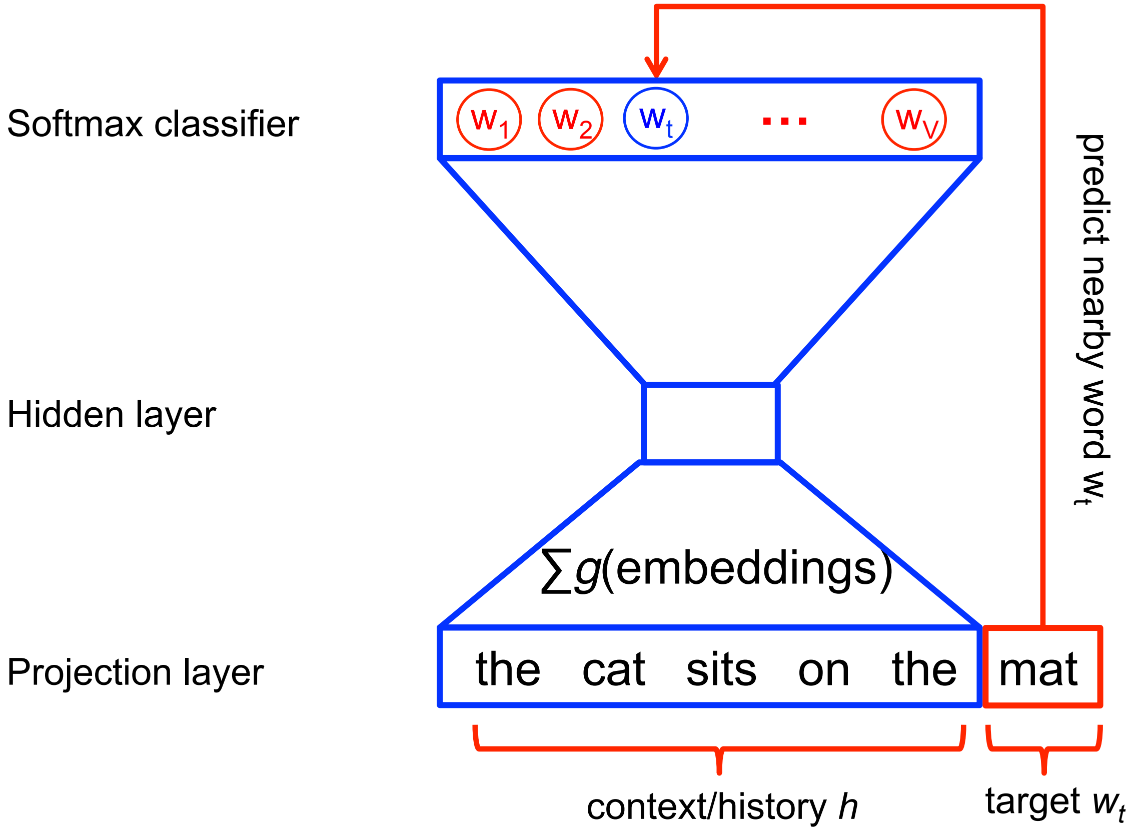
Word Embedding Models

Models differ based on:

1. How they compute the context **h**
   - positional / convolutional / bag-of-words
2. How they map context to target $w_t = f(\mathbf{h})$
   - linear, bilinear, non-linear
3. How they measure $loss_R(w_t, f(\mathbf{h}))$ and how they're trained
   - **language models (NLL, …)**
   - **word embeddings**: negative sampling (CBOW/ Skipgram), sampled rank loss (Collobert+Weston), squared-error (LSA)



Learning Word Embeddings: Pre-2008



$$Pr(w_t) = \frac{\text{score}(w_t; \theta)}{\sum_{j=1}^{V} \text{score}(w_j; \theta)}$$

(Bengio *et al.*, 2003)

Advances in Learning Word Embeddings

Not interested in language modelling (for which we need normalized probabilities), so we don't need the expensive softmax. Can use much faster

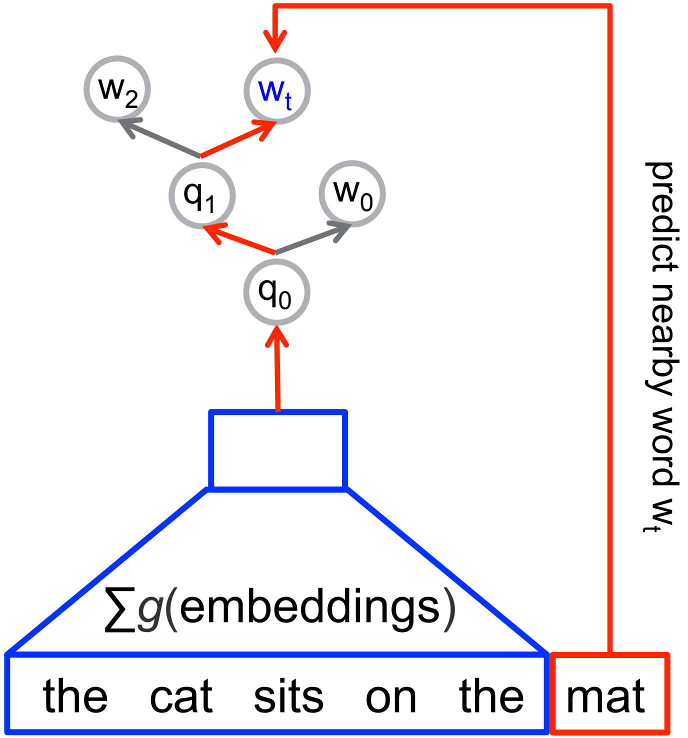
- **hierarchical softmax** (Morin + Bengio, 2005),
- **sampled rank loss** (Collobert + Weston, 2008)**,**
- **noise-contrastive estimation** (Mnih + Teh, 2012)
- **negative sampling** (Mikolov et al., 2013)

Learning Word Embeddings: **Hierarchical Softmax**

Hierarchical
Softmax classifier

Hidden layer

Projection layer

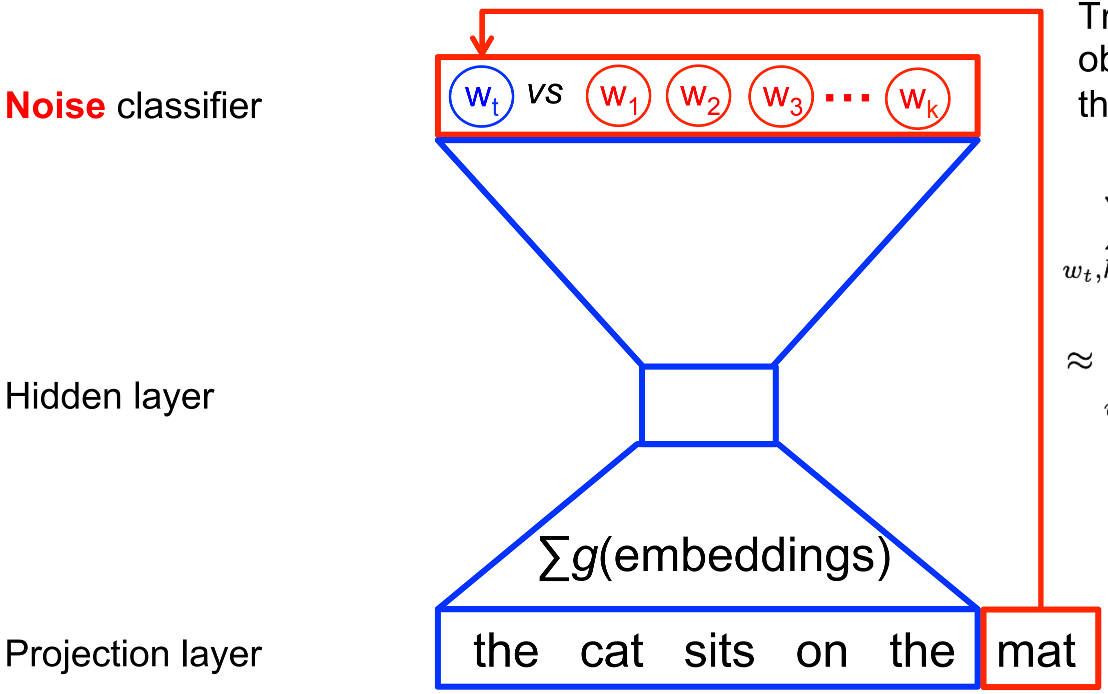


$$P(w_t|h) = \prod_{i \in L(w_t)} P(q_i|h)$$

Significant savings since |L(w)| << V

(Morin + Bengio, 2005;
Mikolov *et al.*, 2013)

Learning Word Embeddings: **Sampled rank loss**



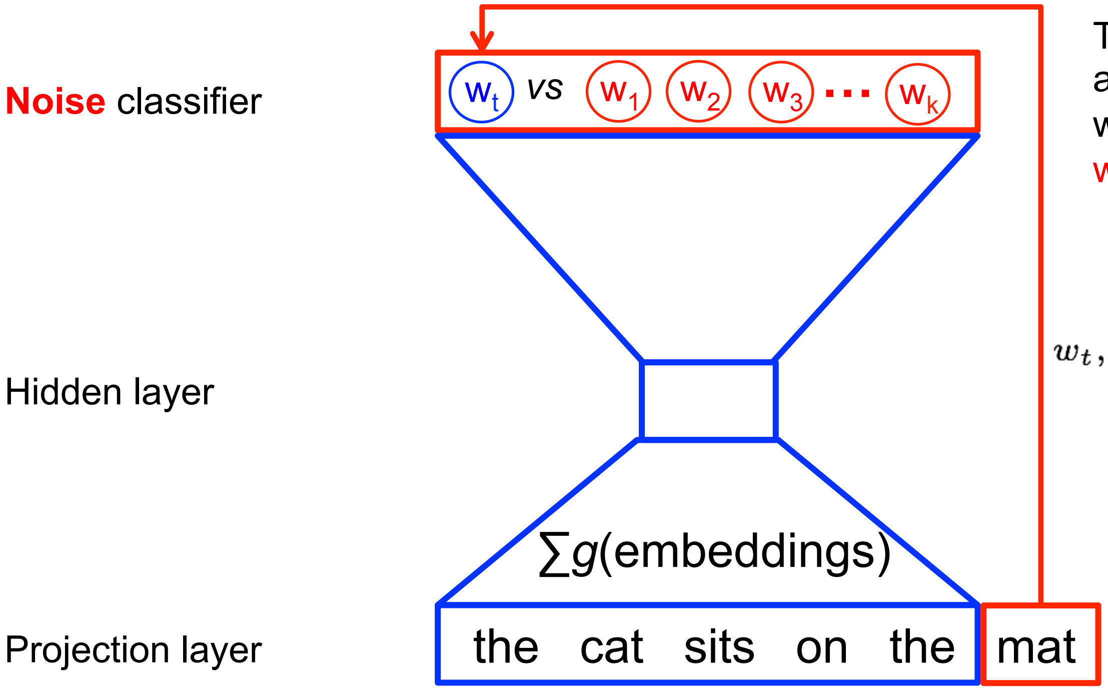
Train a **non-probabilistic** model to **rank** an observed word $w_t \sim P_{data}$ some margin higher than k (<<V) sampled noise words $w_{noise} \sim P_{noise}$

$$\sum_{w_t, h \sim \text{data}} \sum_{j=1}^{V} \max\{0, 1 - (s_\theta(w_t; h) - s_\theta(w_j, h))\}$$

$$\approx \sum_{w_t, h \sim \text{data}} \sum_{j \sim P(w)}^{k} \max\{0, 1 - (\underbrace{s_\theta(w_t; h)}_{\text{increase}} - \underbrace{s_\theta(w_j, h)}_{\text{decrease}})\}$$

(Collobert + Weston, 2008)

Learning Word Embeddings: **Noise-contrastive Estimation**



Train a **probabilistic** model P(w|h) to be able to discriminate an observed nearby word $w_t \sim P_{data}$ from sampled noise words $w_{noise} \sim P_{noise}$

$$\sum_{w_t, h \sim P_{\mathrm{data}}} P(w_t|h) + \sum_{j \sim P_{\mathrm{noise}}} (1 - P(w_j|h))$$

(Mnih + Teh, 2012)

Neural Word Embeddings

Why do “similar” words have similar embeddings?


J.R. Firth

$$\text{Citizens of} \left\{ \begin{array}{c} \text{France} \\ \text{Denmark} \\ \ldots \\ \text{Sweden} \end{array} \right\} \text{protested today}$$

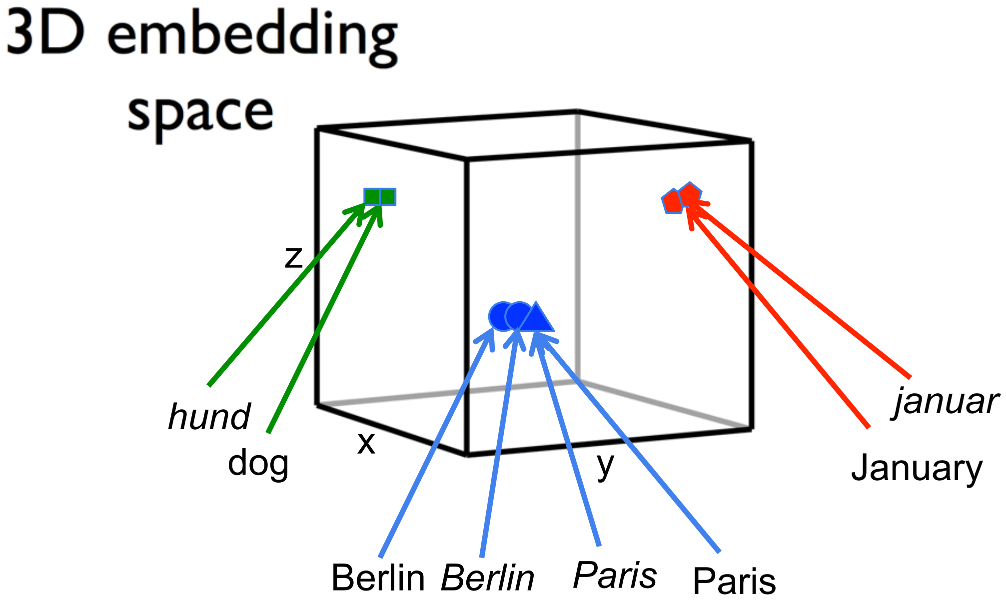
All training objectives have the form:

$$\min_{\mathbf{R}} \sum_i J(w_t^{(i)}, h^{(i)})$$

$$= \min_{\mathbf{R}} \sum_i \text{distance}(w_t^{(i)}, h^{(i)})$$

I.e. for a fixed context, all distributionally similar words will get updated towards a common point.

Cross-lingual Word Embeddings



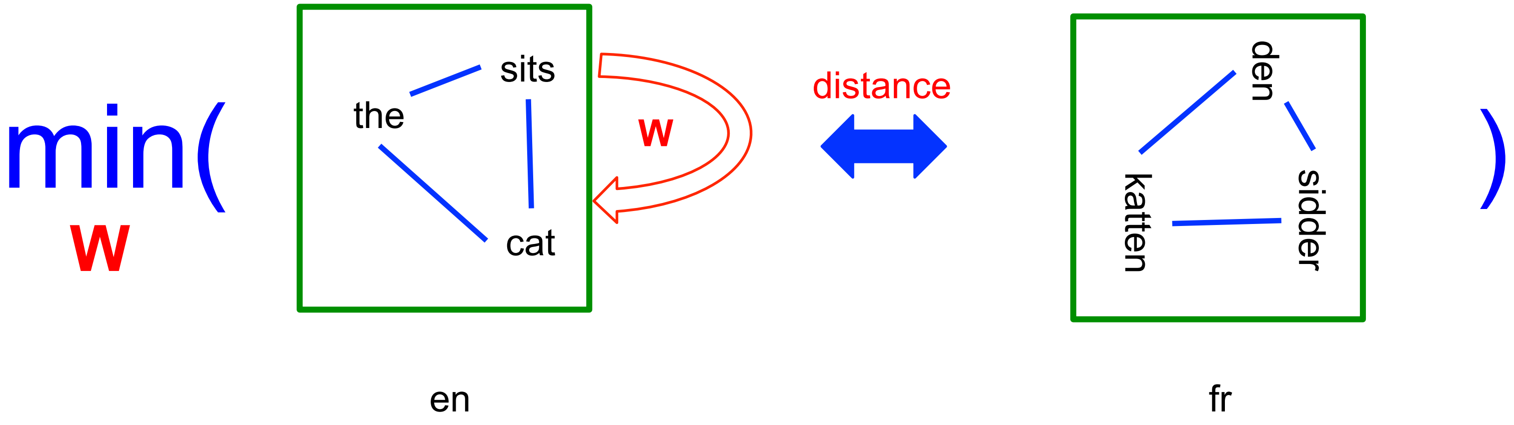
We want to learn an alignment between the two embedding spaces s.t. translation pairs are close.

Learning **Cross-lingual** Word Embeddings: Approaches

1. Align pre-trained embeddings (**offline**)
2. Jointly learn and align embeddings (**online**) using parallel-only data
3. Jointly learn and align embeddings (**online**) using monolingual **and** parallel data

Learning **Cross-lingual** Word Embeddings I

Offline methods: “Translation Matrix”



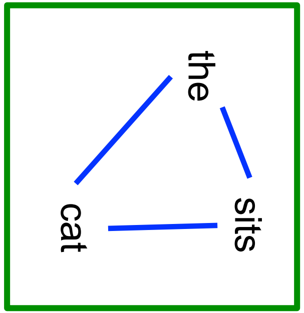
Learn **W** to transform the pre-trained English embeddings into a space where the distance between a word and its translation-pair is minimized

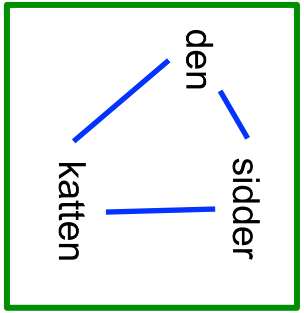
$$\min_{\mathbf{W}} ||\mathbf{R}^{en}\mathbf{W} - \mathbf{R}^{fr}||^2$$

(Mikolov et al., 2013)

Learning **Cross-lingual** Word Embeddings I

Offline methods





en x **W**

fr

Transformed embeddings

Learn **W** to transform the pre-trained English embeddings into a space where the distance between a word and its translation-pair is minimized

$$\min_{\mathbf{W}} ||\mathbf{R}^{en}\mathbf{W} - \mathbf{R}^{fr}||^2$$

Can also learn a *separate* **W** for *en* and *fr* *using* ***Multilingual CCA*** (Faruqui et al, 2014)

(Mikolov et al., 2013)

Learning **Cross-lingual** Word Embeddings II

Parallel-only methods

$$\min_{R}\Big(\quad \text{En parallel} \xleftrightarrow{\text{distance}} \text{Fr parallel} \quad \Big)$$

En parallel

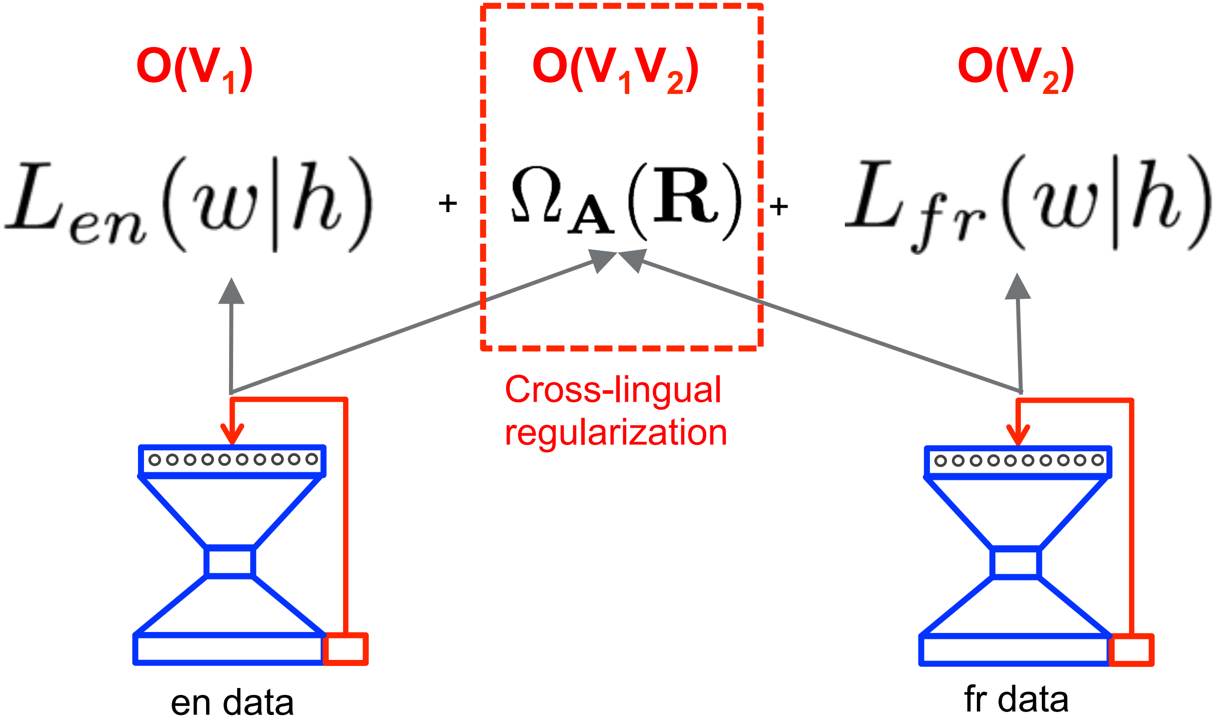
distance

Fr parallel

Bilingual Auto-encoders
(Chandar *et al.*, 2013)
BiCVM
(Hermann *et al.*, 2014)

Learning **Cross-lingual** Word Embeddings III

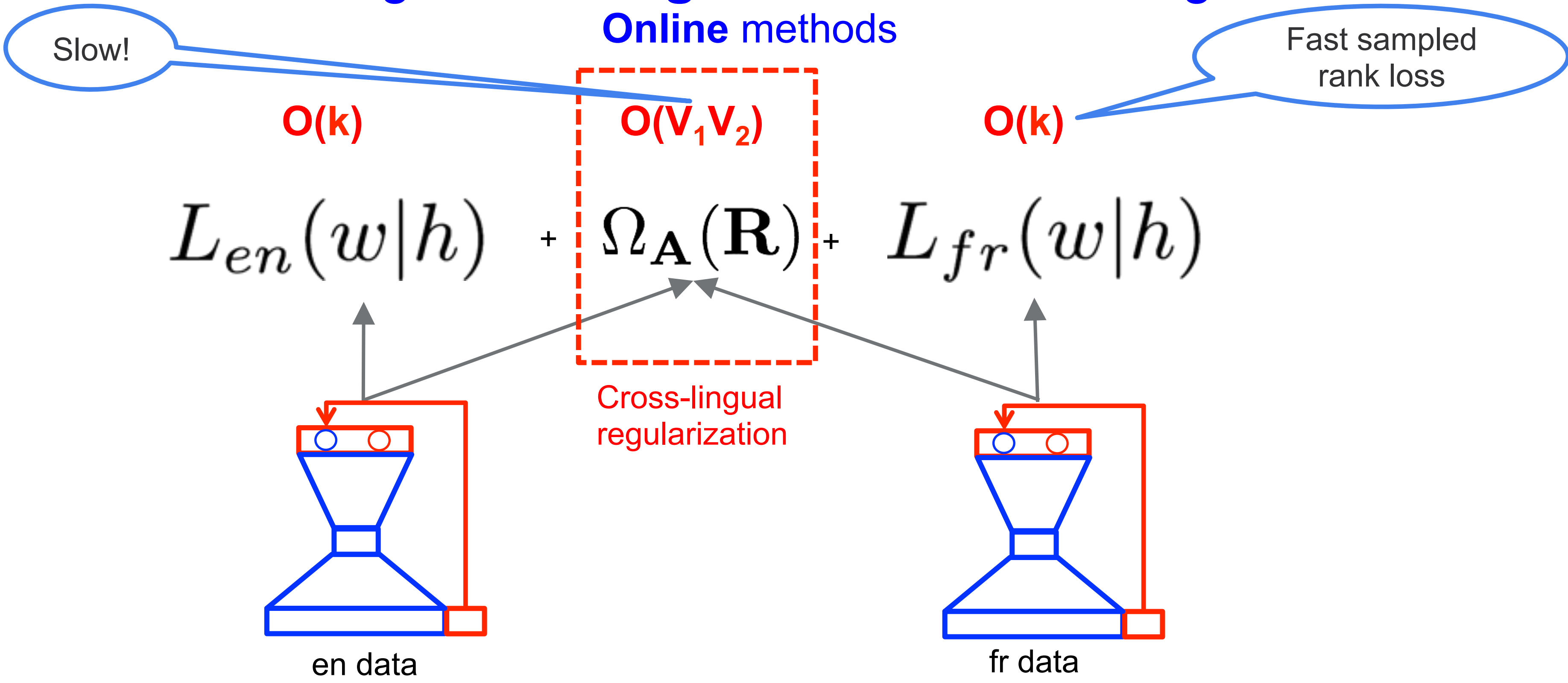
Online methods



(Klementiev et al., 2012)

Learning **Cross-lingual** Word Embeddings III

Online methods



(Zhu et al., 2013)

Multilingual distributed feature learning: Trade-offs

	METHOD	PROS	CONS
OFFLINE	Translation Matrix *(Mikolov et al. 2013)*	• FAST • Simple to implement	• Assumes a global, linear, *one-to-one mapping* exists between words in 2 languages. • Requires accurate dictionaries
	Multilingual CCA *(Faruqui et al. 2014)*		

Multilingual distributed feature learning: Trade-offs

<table>
<tr><th></th><th>METHOD</th><th>PROS</th><th>CONS</th></tr>
<tr><td rowspan="2">OFFLINE</td><td>Translation Matrix (Mikolov et al. 2013)</td><td rowspan="2">• FAST
• Simple to implement</td><td rowspan="2">• Assumes a global, linear, one-to-one mapping exists between words in 2 languages.
• Requires accurate dictionaries</td></tr>
<tr><td>Multilingual CCA (Faruqui et al. 2014)</td></tr>
<tr><td rowspan="2">PARALLEL</td><td>Bilingual Auto-encoders (Chandar et al. 2014)</td><td>Simple to implement (?)</td><td rowspan="2">• Bag-of-words models
• Learns more semantic than syntactic features
• Reduced training data
• Big domain bias</td></tr>
<tr><td>BiCVM (Hermann et al., 2014)</td><td>Allows arbitrary differentiable sentence composition function</td></tr>
</table>

Multilingual distributed feature learning: Trade-offs

	METHOD	PROS	CONS
OFFLINE	Translation Matrix *(Mikolov et al. 2013)*	• FAST • Simple to implement	• Assumes a global, linear, *one-to-one mapping* exists between words in 2 languages. • Requires accurate dictionaries
	Multilingual CCA *(Faruqui et al. 2014)*		
PARALLEL	Bilingual Auto-encoders *(Chandar et al. 2014)*	Simple to implement (?)	• Bag-of-words models • Learns more semantic than syntactic features • Reduced training data • Big *domain bias*
	BiCVM *(Hermann et al., 2014)*	Allows arbitrary differentiable sentence composition function	
ONLINE	*Klementiev et al., 2012*	Can learn fine-grained, cross-lingual syntactic/ semantic features (depends on window-length)	• SLOW • Requires word-alignments (GIZA++/Fastalign)
	Zhu et al., 2013		

Multilingual distributed feature learning: Trade-offs

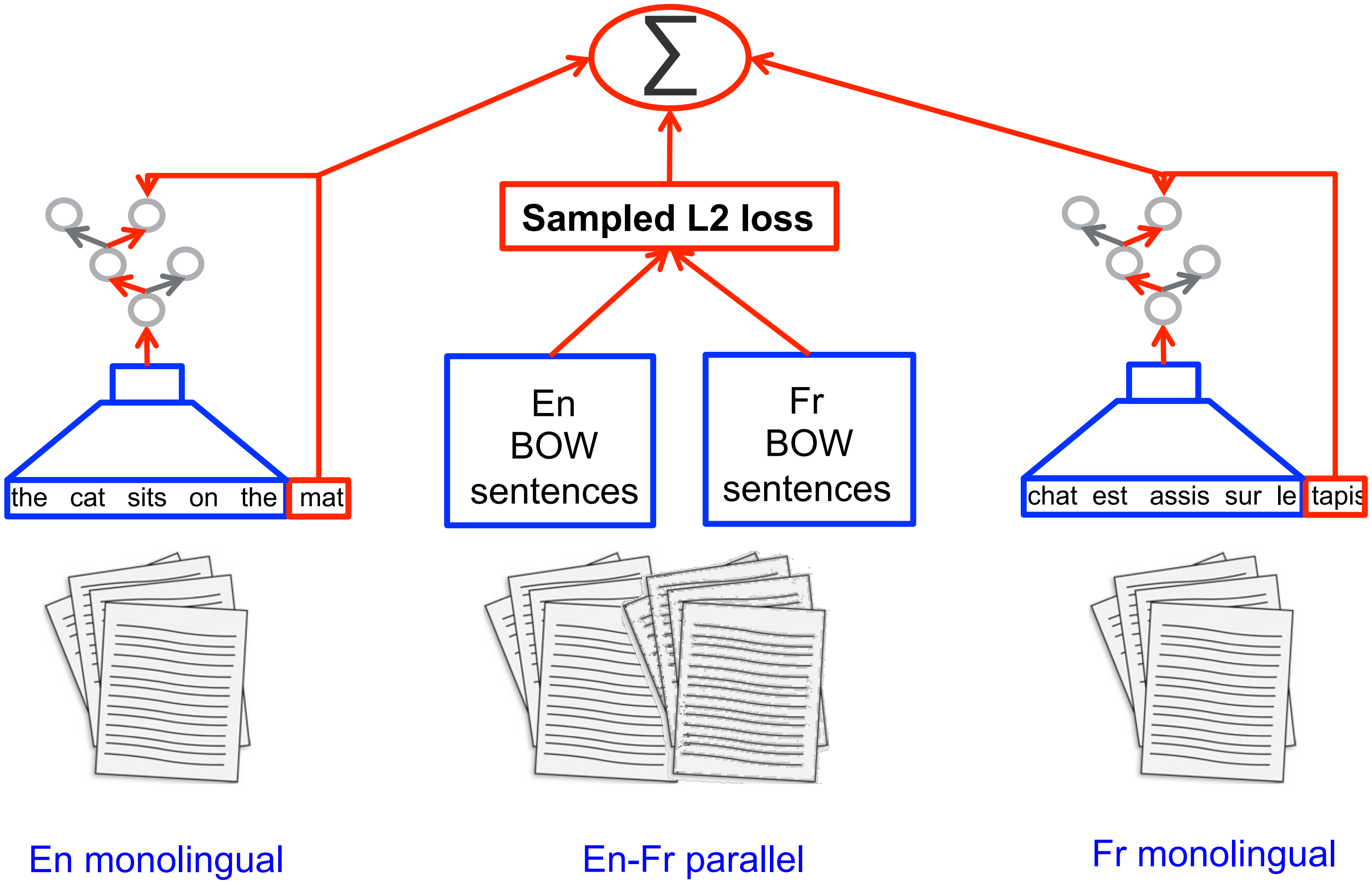
	METHOD	PROS	CONS
OFFLINE	Translation Matrix *(Mikolov et al. 2013)*	• FAST • Simple to implement	• Assumes a global, linear, *one-to-one mapping* exists between words in 2 languages. • Requires accurate dictionaries
	Multilingual CCA *(Faruqui et al. 2014)*		
PARALLEL	Bilingual Auto-encoders *(Chandar et al. 2014)*	Simple to implement (?)	• Bag-of-words models • Learns more semantic than syntactic features • Reduced training data • Big *domain bias*
	BiCVM *(Hermann et al., 2014)*	Allows arbitrary differentiable sentence composition function	
ONLINE	*Klementiev et al., 2012*	Can learn fine-grained, cross-lingual syntactic/ semantic features (depends on window-length)	• SLOW • Requires word-alignments (GIZA++/Fasta
	Zhu et al., 2013		

This work makes multilingual distributed feature learning more efficient for transfer learning and translation.

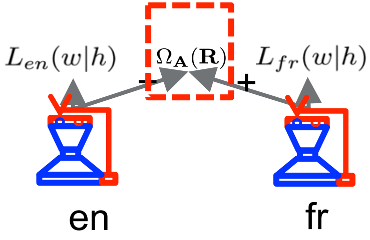
BilBOWA:

Fast Bilingual Bag-Of-Words Embeddings without Alignments

BilBOWA Architecture



The BilBOWA Cross-lingual Objective I



We want to learn similar embeddings for translation pairs. The **exact** cross-lingual objective to minimize is the weighted sum over all distances of word-pairs:

$$\Omega_{\mathbf{A}}(\mathbf{R}) = \sum_{w_i \in V_{\mathbf{en}}} \sum_{w_j \in V_{\mathbf{fr}}} a_{ij} ||\mathbf{R}^e_{[i,:]} - \mathbf{R}^f_{[j,:]}||^2$$

Alignment "score"

Main contribution: We **approximate** this by sampling parallel sentences.

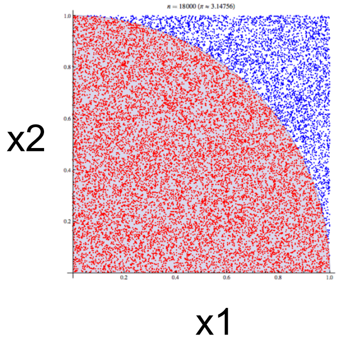
Quick Aside: Monte Carlo integration

$$\mathop{\mathbb{E}}_{p(x)} f(x)$$

$$= \int_x p(x) f(x) \text{ (Continuous)}$$

$$= \sum_{x_i} Pr(x_i) f(x_i) \text{ (Discrete)}$$

$$\approx 1/N \sum_{\color{red}{x \sim p(x)}} f(\color{red}{x})$$



The BilBOWA Cross-lingual Objective II

$$\Omega_{\mathbf{A}}(\mathbf{R}) = \sum_{i,j} a_{ij} ||\mathbf{R}^e_{[i,:]} - \mathbf{R}^f_{[j,:]}||^2$$

$$= \mathop{\mathbb{E}}_{(i,j)\sim P(w^e,w^f)} \left[||\mathbf{R}^e_{[i,:]} - \mathbf{R}^f_{[j,:]}||^2 \right]$$

P(w^e*, w*^{f}*)* is the distribution of en-fr **word alignments**

$$\approx \frac{1}{S} \sum_{s \in S} \frac{1}{mn} \sum_{(i,j) \in s} ||\mathbf{R}^e_{[i,:]} - \mathbf{R}^f_{[j,:]}||^2$$

Assume *P(w*^{e}*, w*^{f}*)* is uniform. m/n length of en/fr sentence.

Now we set S = 1 at each time step *t*:

$$\Omega^{(t)}_{\mathbf{A}}(\mathbf{R}) = || \underbrace{\frac{1}{m} \sum_{i \in s_e} \mathbf{R}^e_{[i,:]}}_{\text{mean en sentence-vector}} - \underbrace{\frac{1}{n} \sum_{j \in s_f} \mathbf{R}^f_{[j,:]}}_{\text{mean fr sentence-vector}} ||^2$$

mean **en** sentence-vector mean **fr** sentence-vector

Implementation Details: Open-source

Implemented in C (part of `word2vec`, soon). Multi-threaded: One thread per language (monolingual), and one additional thread per language-pair (cross-lingual) (i.e. **asynchronous SGD**)

Runs at ~10-50K words per second on MBP.

Can process 500M words (monolingual) and 45M words (parallel, recycles) in about 2.5h on my MBP.

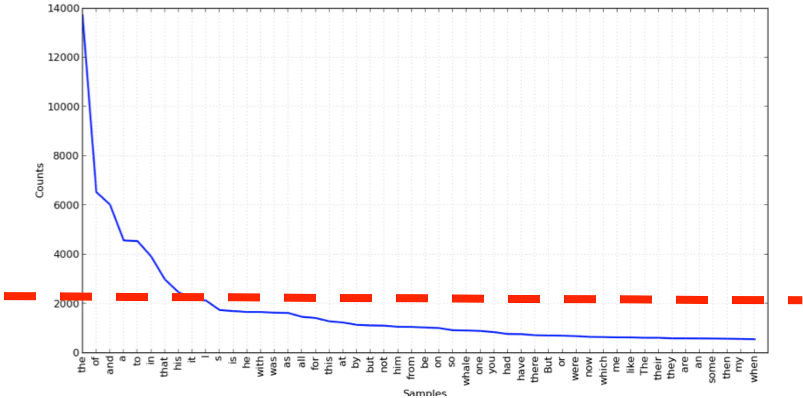
Cross-lingual subsampling for better results

At training step *t*, draw a random number u ~ U[0,1]. Then:

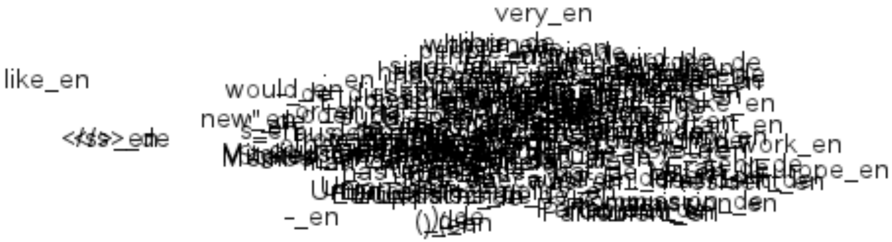
$$\Omega_{\mathbf{A}}^{(t)}(\mathbf{R}) = || \frac{1}{m} \sum_{i \in s_e} \mathbb{1}_{u<f(w_i)} \mathbf{R}^{e}_{[i,:]} - \frac{1}{n} \sum_{j \in s_f} \mathbb{1}_{u<f(w_j)} \mathbf{R}^{f}_{[j,:]} ||^2$$

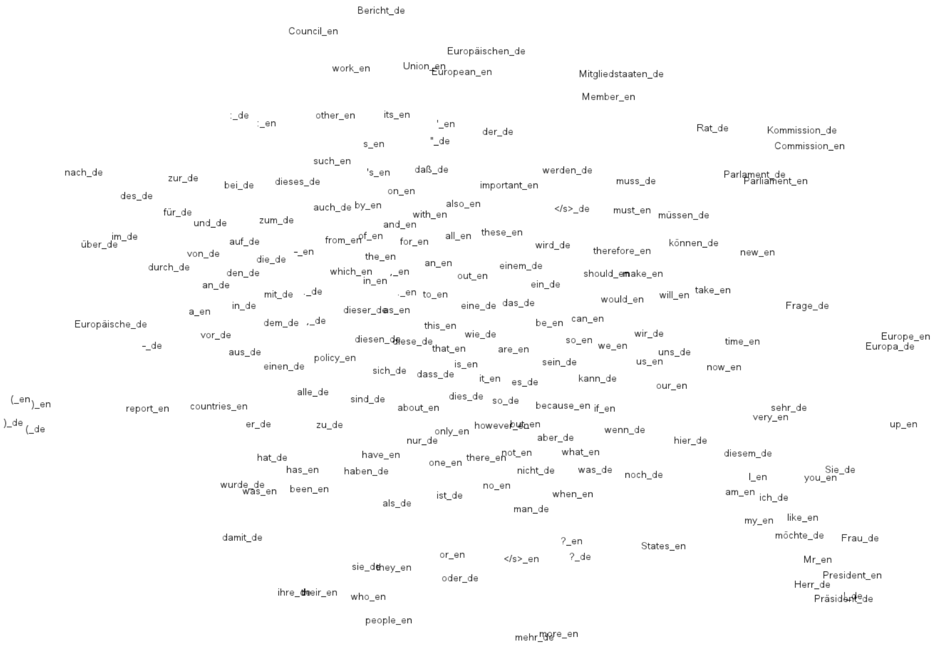
Want to estimate **alignment** statistics P(e,f).
Skewed at the sentence-level by (unconditional) unigram word frequencies.

Simple solution:
Subsample frequent words to flatten the distribution!



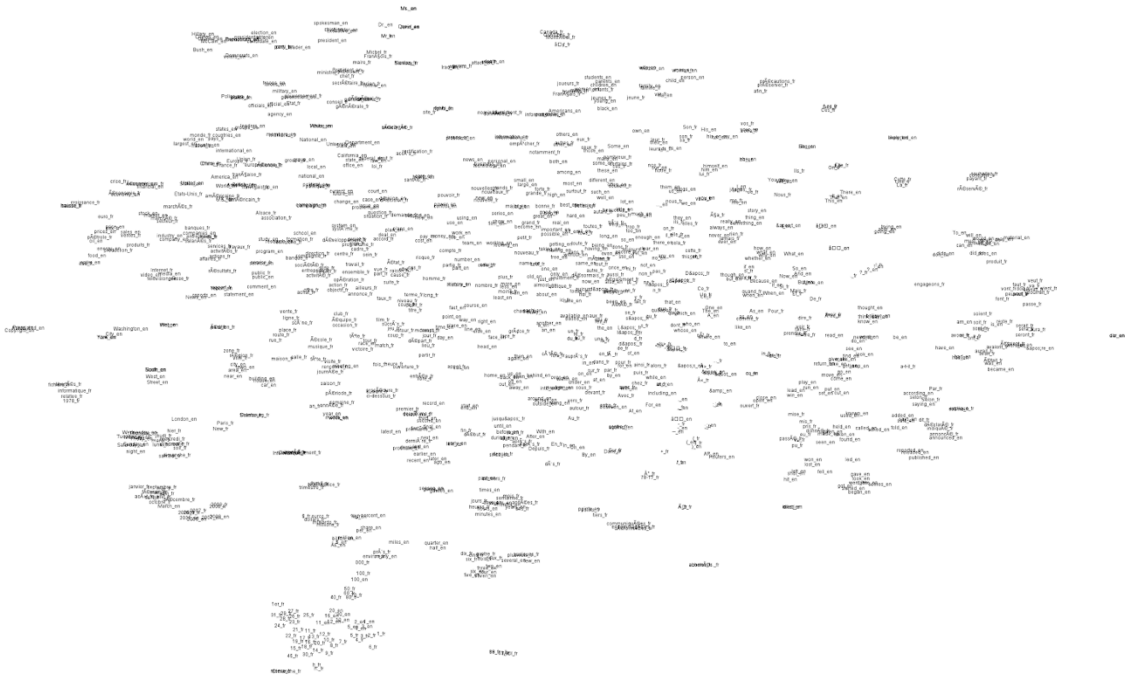
Cross-lingual subsampling for better results



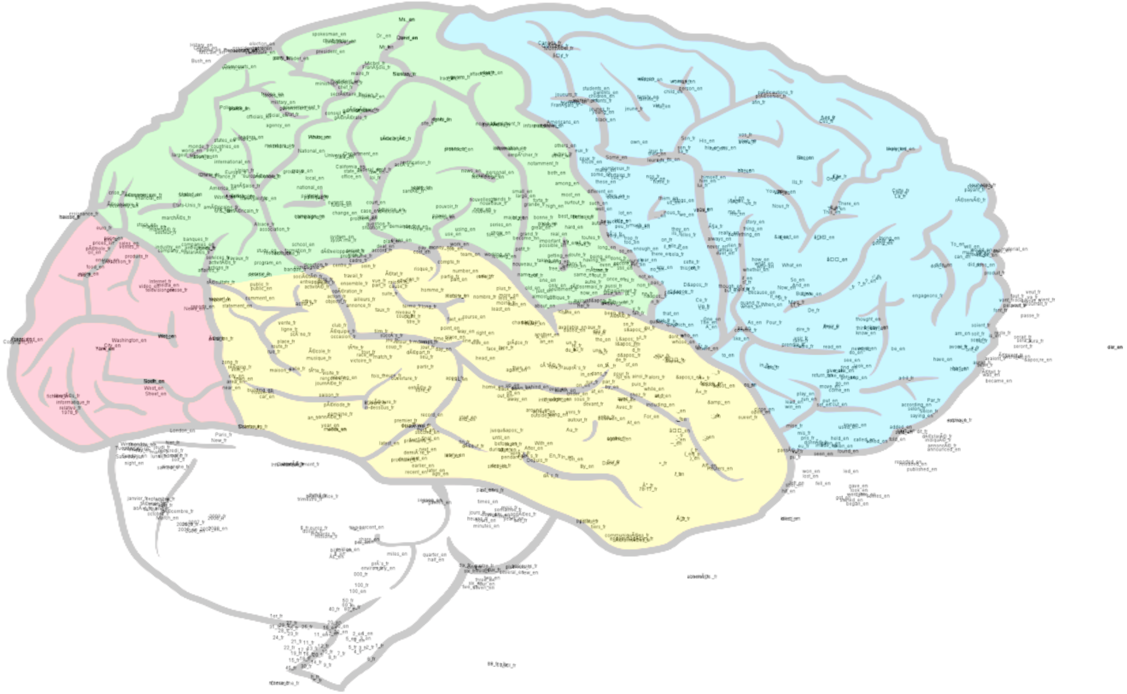


Without SS **With SS**

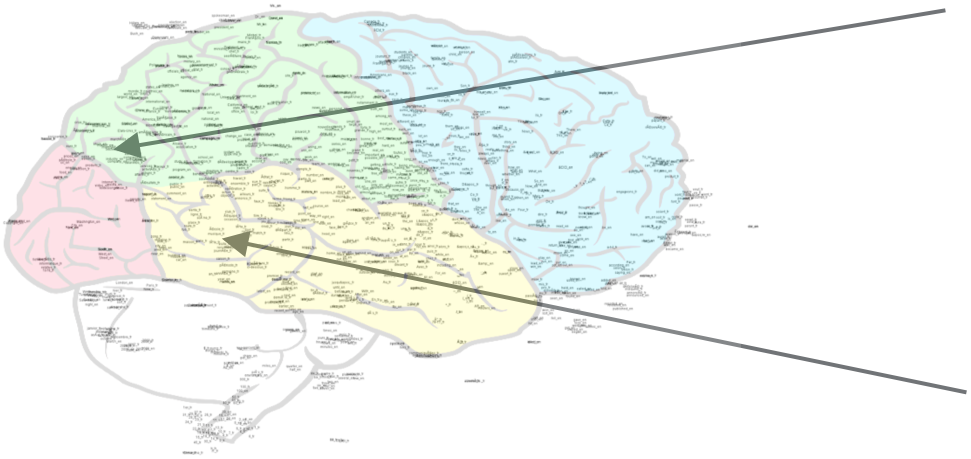
Qualitative Analysis: *en-fr* t-SNEs I

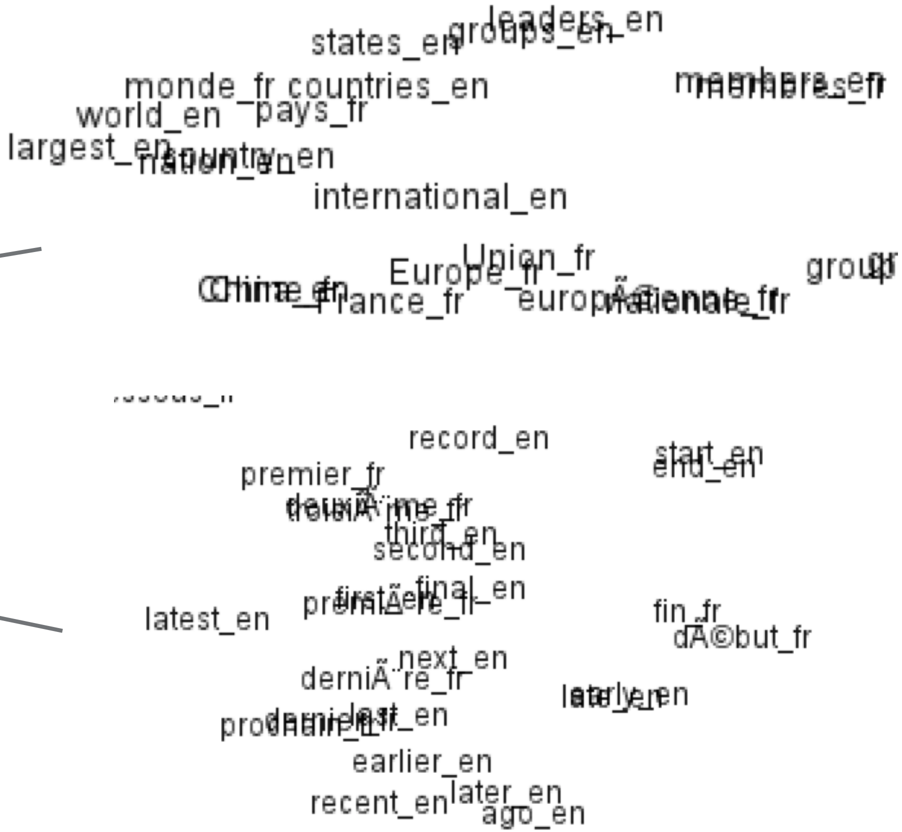


Qualitative Analysis: *en-fr* t-SNEs I

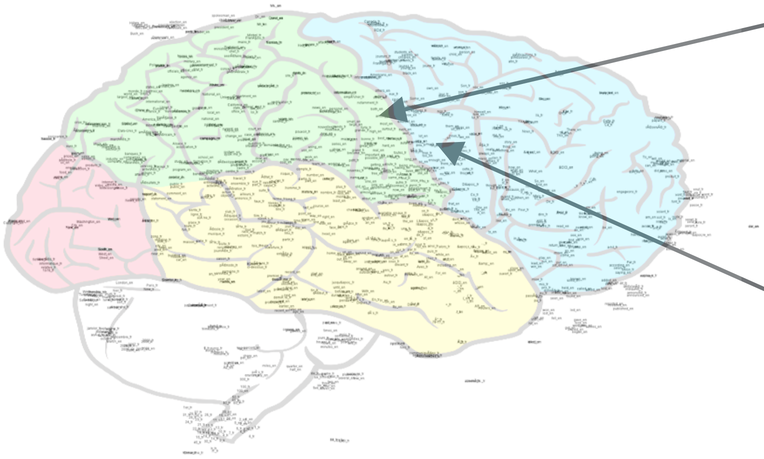


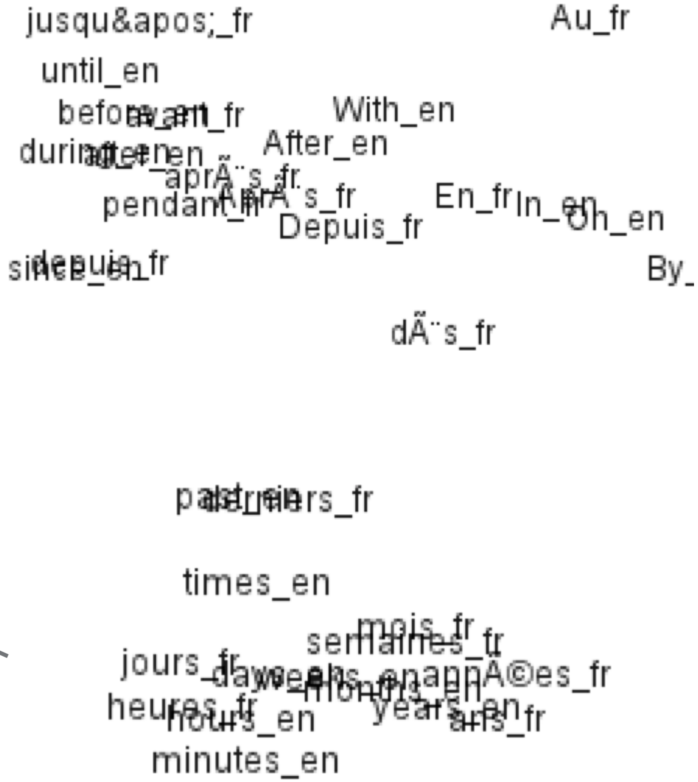
Qualitative Analysis: *en-fr*



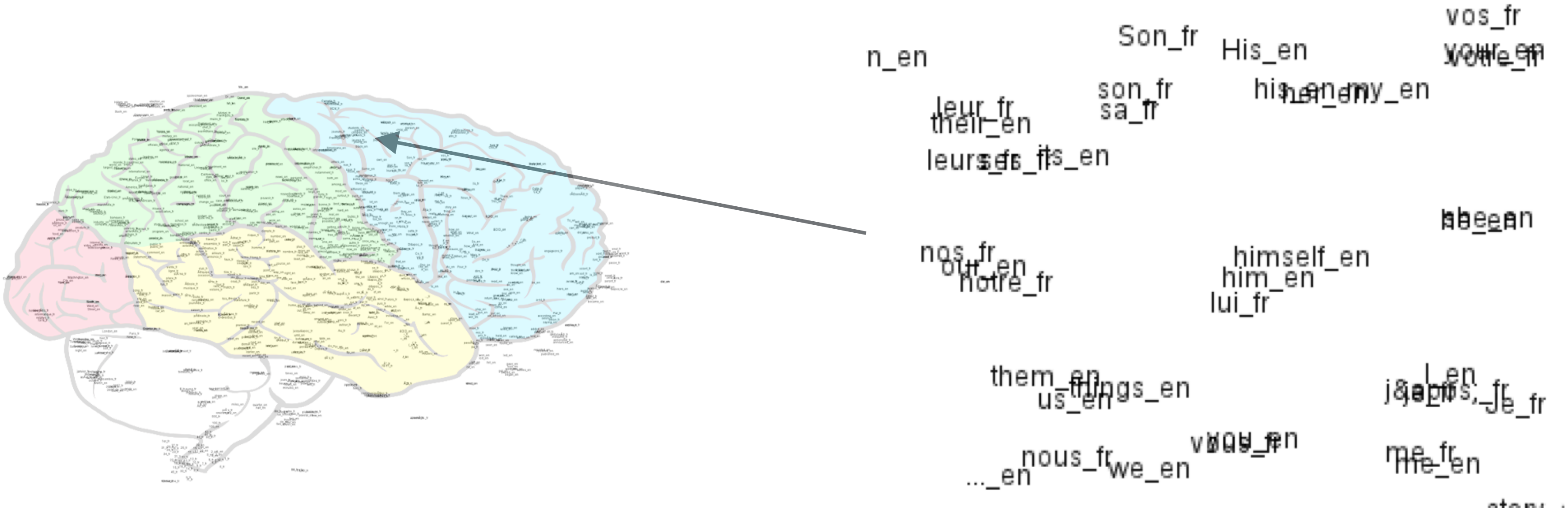


Qualitative Analysis: *en-fr*

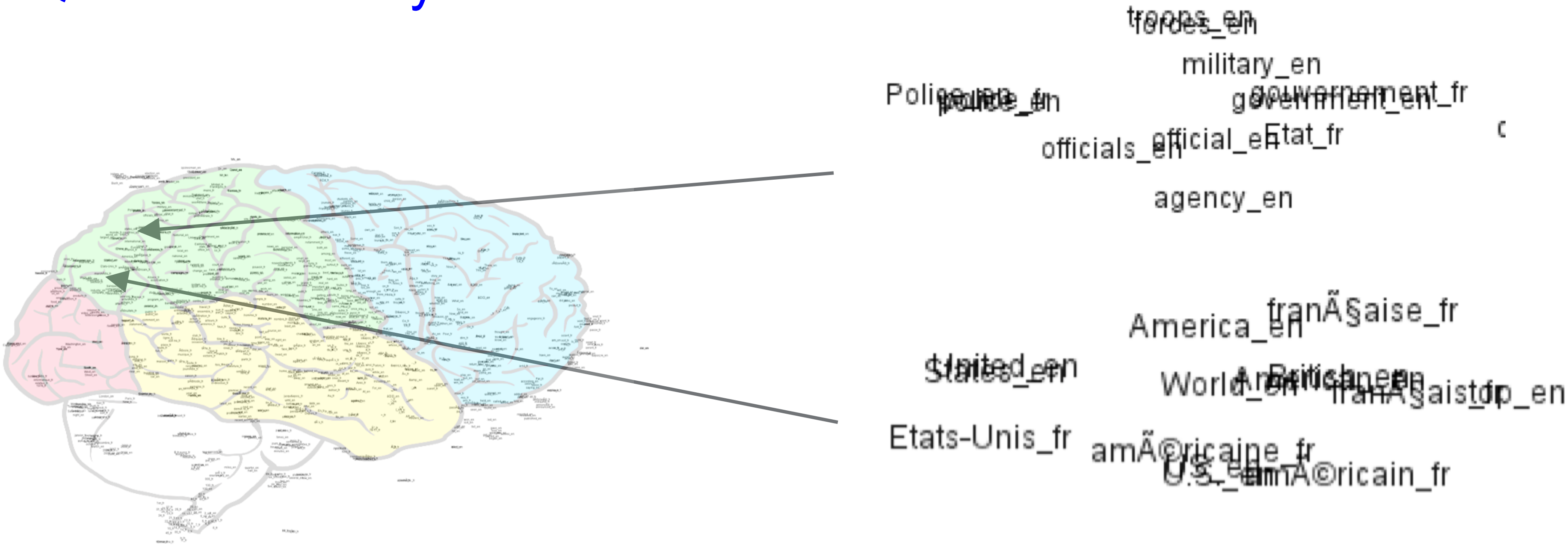




Qualitative Analysis: *en-fr*



Qualitative Analysis: *en-fr*



Experiments: *En-De* Cross-lingual Document Classification

Exact replication (obtained from the authors) of *Klementiev et al.*'s cross-language document classification setup:

Goal: Classify documents in target language using only labelled documents in source language.

Data: English-German RCV1 data (5K test, 100 - 10K training, 1K validation)

4 Labels:

- CCAT (Corporate/Industrial),
- ECAT (Economics),
- GCAT (Government/Social), and
- MCAT (Markets)

Results: English-German Document Classification

	en2de	***de2en***	***Training Size***	***Training Time (min)***
Majority class	46.8	46.8	-	-
Glossed	65.1	68.6	-	-
MT	68.1	67.4	-	-
Klementiev et al.	77.6	71.1	50M words	14,400 (10 days)

Results: English-German Document Classification

	en2de	***de2en***	***Training Size***	***Training Time (min)***
Majority class	46.8	46.8	-	-
Glossed	65.1	68.6	-	-
MT	68.1	67.4	-	-
Klementiev et al.	77.6	71.1	50M words	14,400 (10 days)
Bilingual Autoencoders	**91.8**	72.8	50M words	4,800 (3.5 days)
BiCVM	83.7	71.4	50M words	15
BilBOWA (this work)	86.5	**75**	50M words	**6**

Experiments: WMT11 *English*-*Spanish* Translation

- Trained BilBOWA model on En-Es Wikipedia/Europarl data.
  - Vocabulary = 200K
  - Embedding dimension = 40,
  - Crosslingual **λ**-weight in {0.1, 1.0, 10.0, 100.0}

- Exact replica of *(Mikolov, Le, Sutskever, 2013):*
  - Evaluated on WMT11 lexicon, translated using GTranslate
  - Top 5K-6K words as test set

Experiments: WMT11 ***English→Spanish*** Translation

	Dimension	***Prec@1***	***Prec@5***	***% Coverage***
Edit distance	-	13	24	92.9
Word co-occurrence	-	19	30	92.9
Translation Matrix	*300-800*	33	51	92.9
BilBOWA (This work)	40	**39 (+6%)**	**51**	92.7

Experiments: WMT11 ***Spanish→English*** Translation

	Dimension	***Prec@1***	***Prec@5***	***% Coverage***
Edit distance	-	18	27	92.9
Word co-occurrence	-	20	30	92.9
Translation Matrix	*300-800*	35	52	92.9
BilBOWA (This work)	40	**44 (+11%)**	**55 (+3%)**	92.7

BARISTA:

Bilingual Adaptive Reshuffling *with* Individual Stochastic Alternatives

Motivation

Word embedding models learn to predict targets from contexts by clustering similar words into soft (distributional) equivalence classes

For some tasks, we may easily obtain the desired equivalence classes:

- **POS**: Wiktionary
- **Super-sense** (SuS) tagging: WordNet
- **Translation**: Google Translate / dictionaries

BARISTA embeds additional task-specific semantic information by corrupting the training data according to known equivalences C(w).

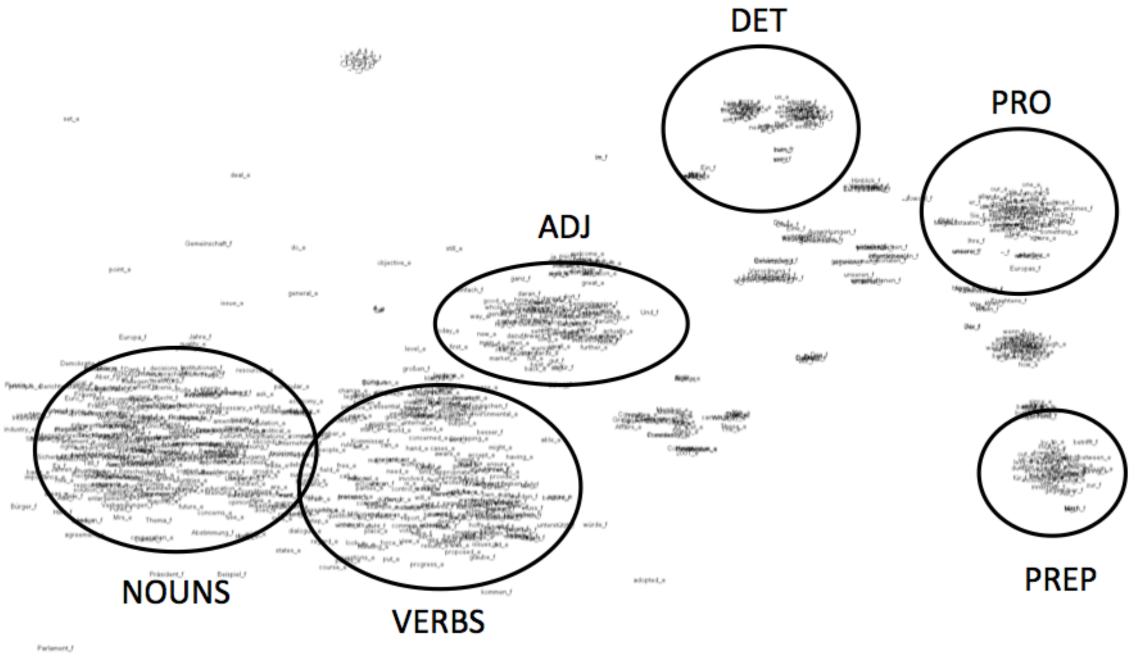
BARISTA Algorithm

```
1. Shuffle D_en & D_fr -> D
2. For w in D:
3.   If w in C then w’ ~ C(w) else w’ = w
4.   D’ += w’
4. Train off-the-shelf embedding model on D’
```

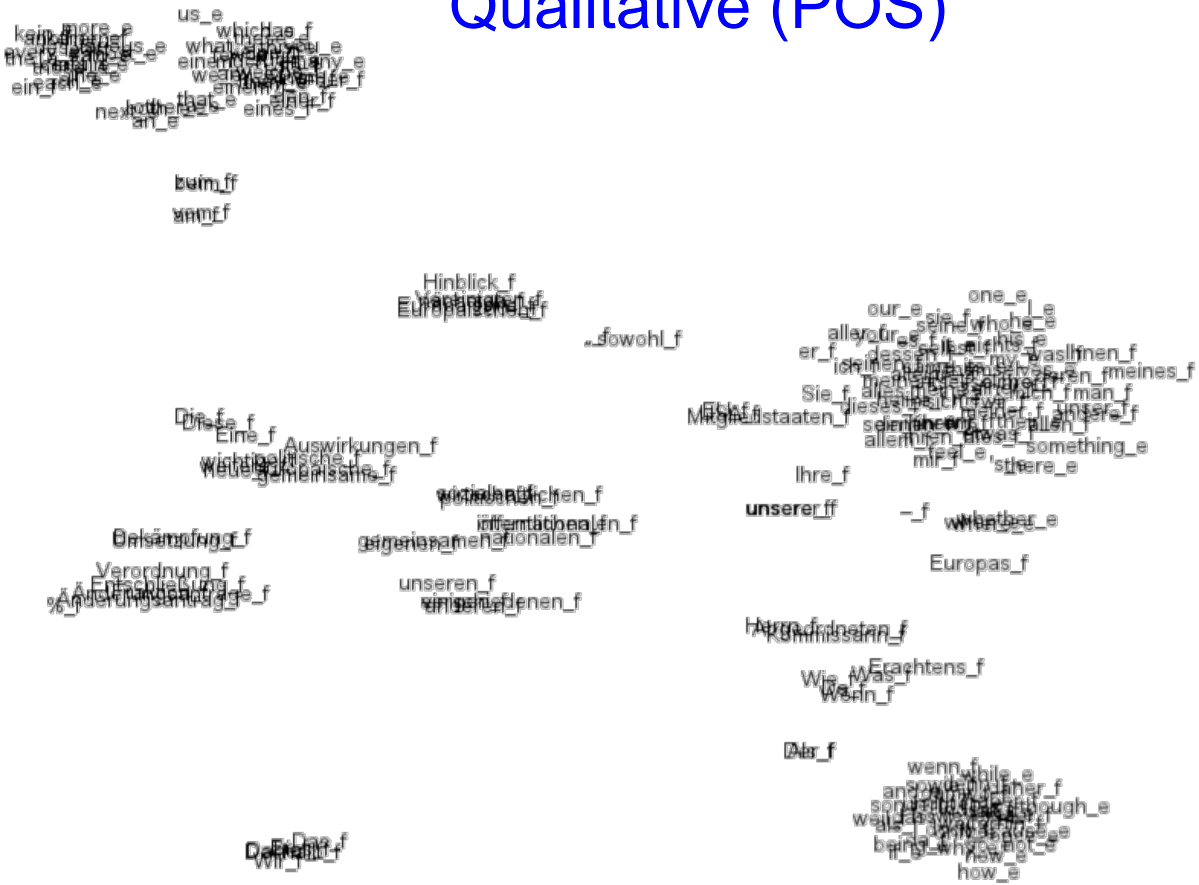
For example: *“we build the house”:*

1. *we build la voiture / they run la house* ***(POS)***
2. *we construire the maison / nous build la house* ***(Translations)***

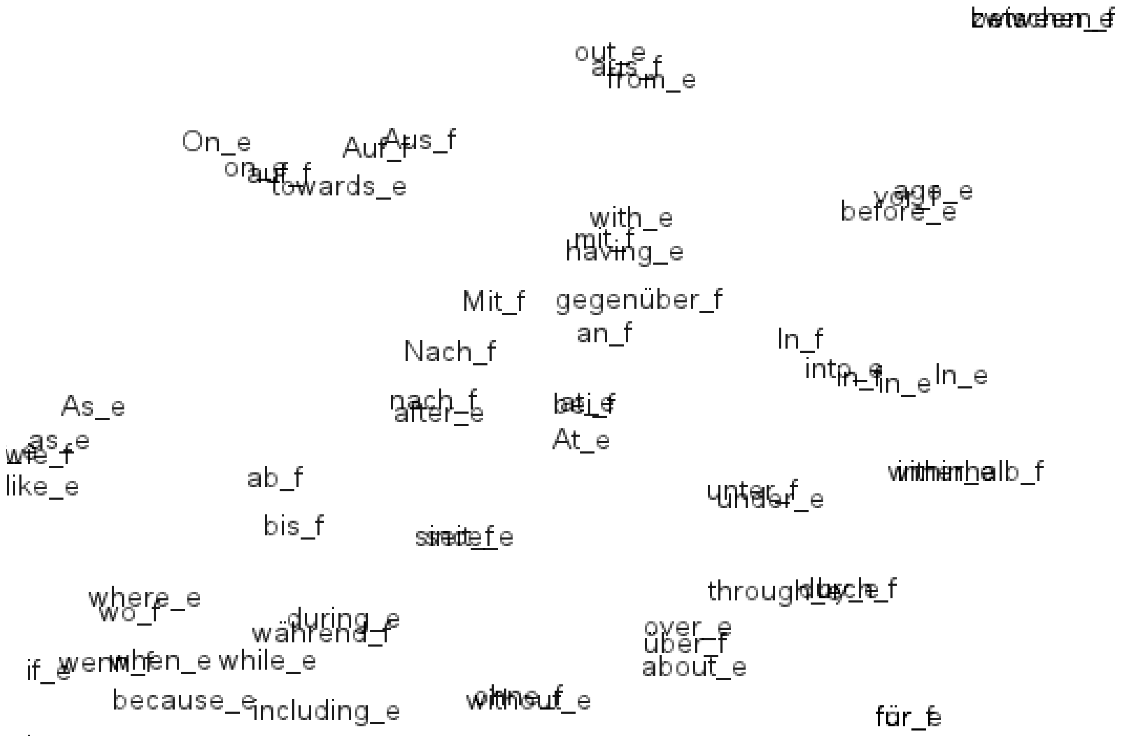
Qualitative (POS)



Qualitative (POS)

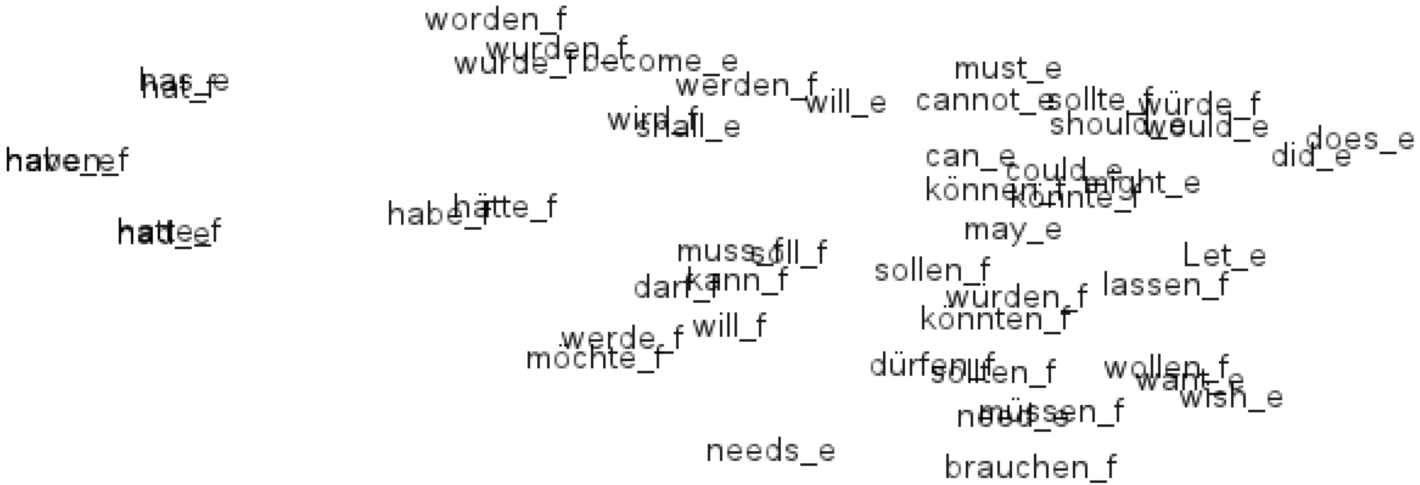


Qualitative (Translations)



Prepositions

Qualitative (Translations)



Modals

Cross-lingual POS tagging

Language	Baseline	Random	Klmtv	POS-50	POS-300	Tr-50	Tr-300	DP	B-K
Spanish	80.6	81.8	79.8	82.4	81	81.6	82.6	**84.2**	80.2
German	80.4	82.7	82.8	81.8	84.1	82.6	**84.8**	82.8	81.3
Danish	63	68.9	-	68.9	72.4	71.8	78.4	**83.2**	69.1
Swedish	71.6	73.7	-	75	76	75.4	77.5	**80.5**	70.1
Italian	80.1	81.3	-	82.1	80.9	82.1	80.7	**86.8**	68.1
Dutch	74.5	77.2	-	78.3	77.4	78.7	**80.3**	79.5	65.1
Portuguese	76.9	78.1	-	77.3	76.1	80.6	80.5	**87.9**	78.4
Avg	75.3	77.7	-	78	78.3	79	80.7	**83.6**	73.2

Cross-lingual SuperSense (SuS) tagging

	Baselines		BARISTA	
	MFS	bl	Tr-300	WN-300
blogs	49.1	46.7	50.5	**61.9**
forum	44.5	41.2	45.0	**53.9**
magazine	46.5	45.2	50.4	**51.5**
newswire	48.4	45.4	52.7	**60.9**
reviews	48.4	44.9	50.4	**55.8**
speech	51.1	48.4	51.5	**58.4**
Avg	48.1	45.4	50.5	**58.3**

Conclusion

I presented BilBOWA, an efficient, bilingual word embedding model with an open-source C implementation (part of word2vec, soon) and Barista, a simple technique for embedding additional task-specific cross-lingual information.

Qualitative experiments on `En-Fr & En-De` show that the learned embeddings capture fine-grained cross-lingual linguistic regularities.

Quantitative results on `Es,De,Da,Sv,It,Nl,Pt` for:

- semantic transfer (document classification, xling SuS-tagging)
- lexical transfer (word-level translation, xling POS-tagging)

Thanks!

Questions / Comments?