Unified and unsupervised bilingual phrase alignment in specialized domain

Jingshu Liu

Nantes Machine Learning Meetup

2019-12-2

Outline

Introduction

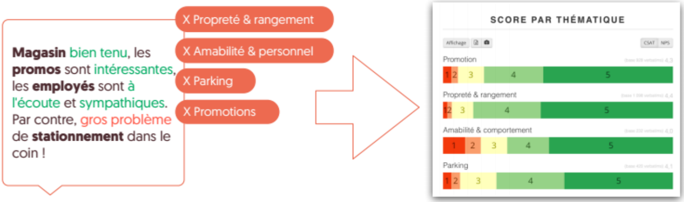
- Word-level representation
- Sequence-level representation
  - Context-independent representation (Compositional)
  - Contextualized representation (Language model)
  - Short sequence representation
- Alignment
  - Word-level alignment (bilingual lexicon induction/extraction)
  - Alignment vs Translation

$$\arg\max_{y} \prod_{t=1}^{T_y} P(y^t | x, y^1, y^2, ..., y^{t-1})$$

$$\arg\max_{y} P(y|x)$$

- Unsupervised learning (particularly in Neural Machine Translation)

Context



- Semantic analysis software editor.
- Key phrase extraction + aspect based sentiment analysis (ABSA).
- International clients.

A unified monolingual phrase representation is the prerequisite for many NLP applications(term classification/clustering). A cross-lingual phrase representation enables an aggregation of all the monolingual data. Besides, it facilitates the transition from one language to another.

Two principles for sequence modeling

- **Compositional principle.** "The meaning of the whole is a function of the meaning of the parts". (*a frying pan*) ⇒ non contextualized
- **Syntactical principle.** "You shall know an object by the company it keeps." (... *a pain in the neck* ...) ⇒ contextualized

What we want at the end...

Align phrases of variable length in specialized domain corpora without cross-lingual information.

- *ankle boot* → *bottine*
- *airflow* → *flux d'air*
- *electric power industry* → 发电业 (one word)

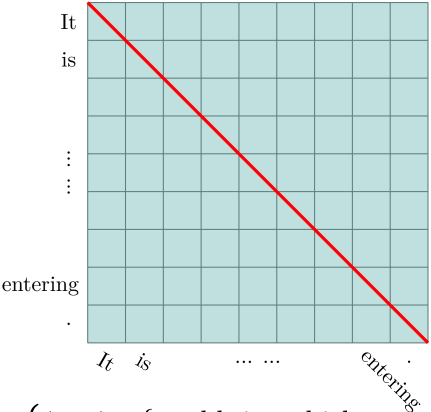
where the phrases are short sequences usually between 1 and 5 words.

Word-level representation

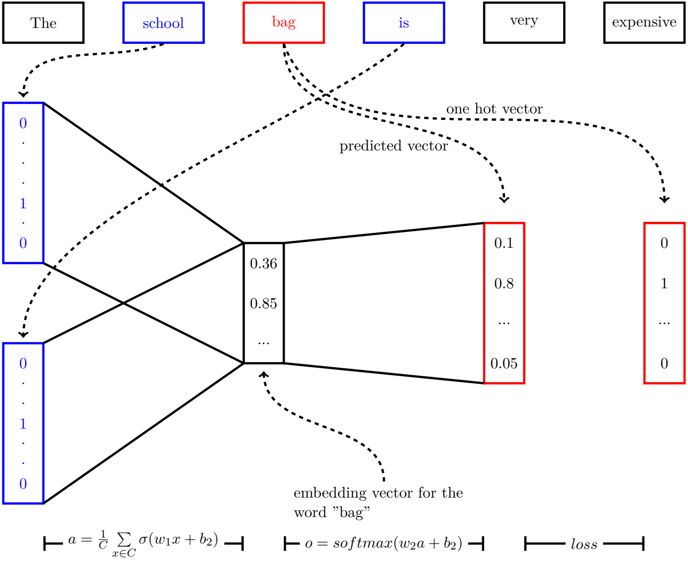
- Word co-occurrence
  - COOC + PMI (No learnable parameters)
  - TF-IDF (No learnable parameters)
  - Glove (With learnable parameters)
- Word-context prediction ($P(w|c_i)$ or $P(c_i|w)$ where each probability for the pair $(w, c_i), c_i \in Context$ is calculated independently)
  - Word2vec (CBOW, Skip-gram)
  - FastText (+subword information)
- From language modeling (next/mask word prediction based), ($P(w_t|w_1, ...w_{t-1})$ or $P(mask_t|w_1, ...w_{t-1}, w_{t+1}, ..., w_n)$ where each context is related.)
  - ELMo (BiLSTM)
  - BERT, XLNET, RoBERTa, XLM, ... (Transformer)

Co-occurrence matrix, window size=3

It is the first vehicle in the world in which passengers pay for their ride upon entering it.



$$v_i^{\text{passengers}} = \begin{cases} 1, & i \in \{\text{world, in, which, pay, for, their}\} \\ 0, & \text{otherwise} \end{cases}$$

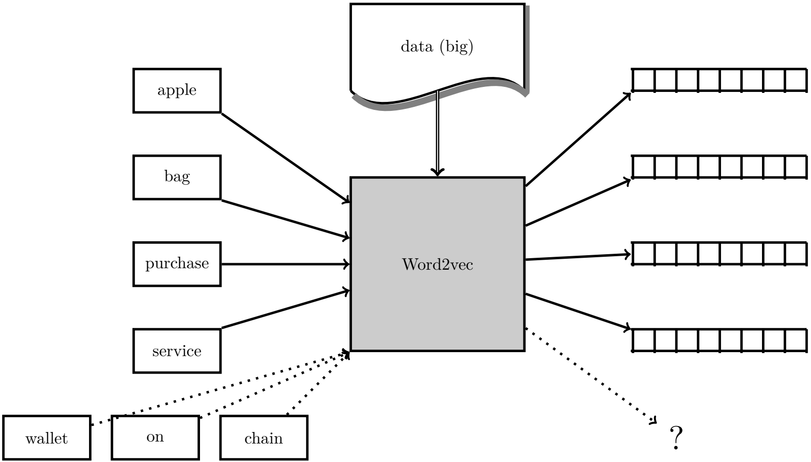


Word2vec (CBOW) with a window size of 1.

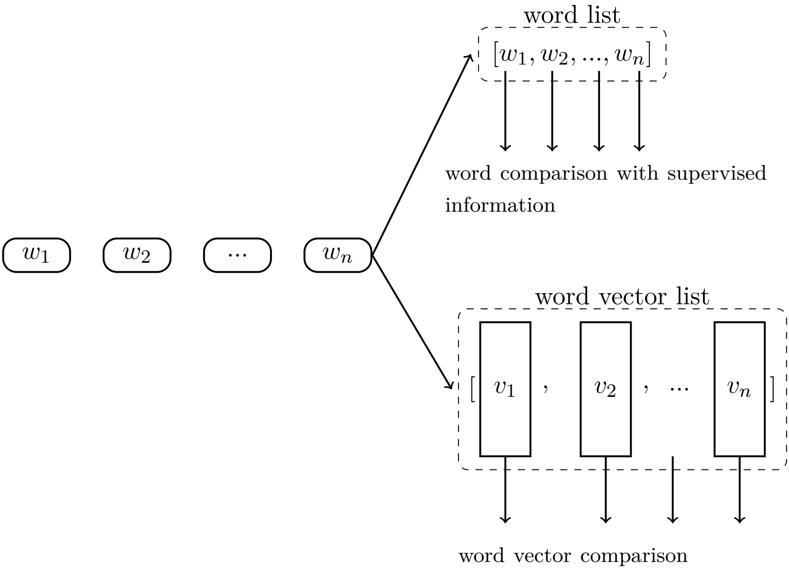
Word-level representation

- Word co-occurrence based ⇒ Explicit, not suitable for large-scale calculation except Glove.
- Word-context prediction based ⇒ Generalized, dense representations suitable for linear transformations.
- From language modeling ⇒ Above + context sensitive (each input has different representations based on the context). But what if one does not have the context for the inference ?

Sequence-level representation



Pure composition



Too naive

- Highly dependent on the task-oriented supervised information.
- Length sensitive.
- Word inner relation ignored. This approach treats separately each word of a multi-word sequence, thus the inner relation between them is completely ignored.

Addition

$$v = \sum_{i=1}^{n} v_i \text{ Or,}$$

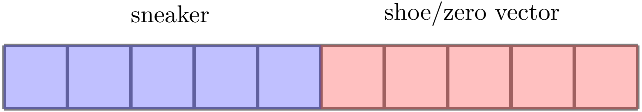
$$v = \frac{\sum\limits_{i=1}^{n} v_i}{n}$$

- Good performance with ability to handle fertility problem.
- Order ignored.
- Uniform weight.

Concatenation

$$v \in \mathbb{R}^{nd}$$

Word order sensitive but variable length phrases are no longer semantically comparable even if we pad them.



$cos(v_{(sneaker,\ shoe)}, v_{(sneaker)})$ will be $\frac{1}{2}$ if we pad the "sneaker" with zeros. This similarity will be probably lower than : $cos(v_{(sneaker,\ shoe)}, v_{(sneaker,\ shop)})$

CBOW/Skip-gram based

- Consider ngrams as one token during the training of CBOW/Skip-gram. (Mikolov, 2013)
  - Works relatively well on idiomatic phrases but less effective on compositional ones.
  - Cannot treat new phrases that were never passed to the training.
- Extended Skip-gram with negative sampling. (Artetxe, 2018) The idea is to update all the ngram vectors in addition to the single word vectors with the same context.

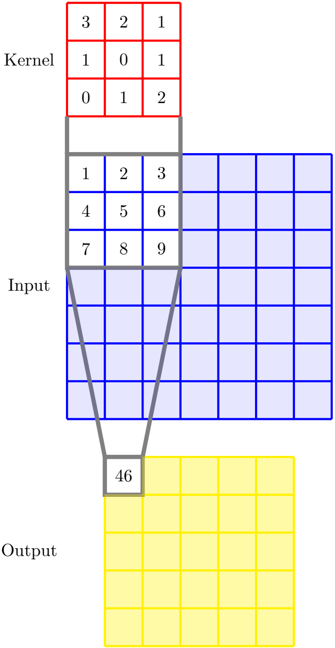
$$E = -\log \sigma(s_{w_O}^T h) - \sum_{w_k \in \mathcal{W}_{neg}} \log \sigma(-s_{w_k}^T h)$$

  where h is the output of the hidden layer for the input word and each ngram token in the same window. s is the output of the hidden layer for the unigram context $c \in w_O \cup \mathcal{W}_{neg}$.
  - Better performance for compositional phrases yet new phrases remains unmanageable.

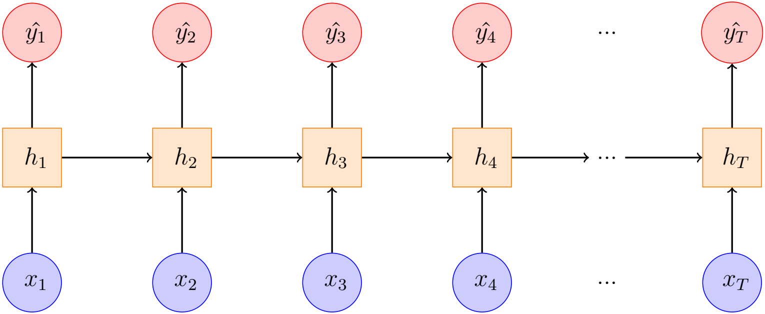
Traditional NN for encoding sequences

- *Convolutional Neural Network* (CNN)
- *Recurrent Neural Network* (denoted by RctNN in our context)
- *Long Short-Term Memory* (LSTM) and *Gated Recurrent Unit* (GRU)
- *Recursive Neural Network* (denoted by RNN in our context)

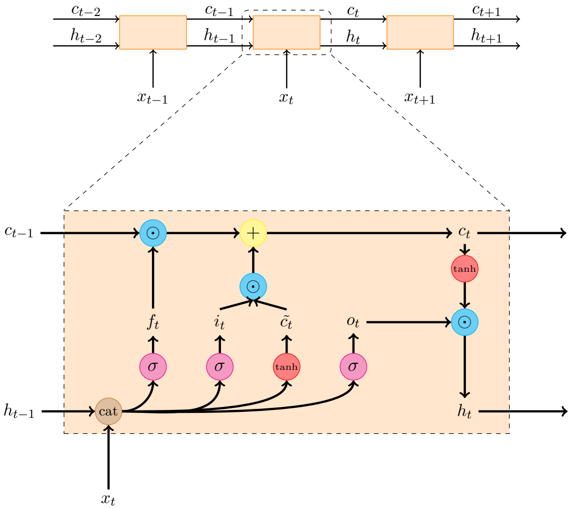
CNN



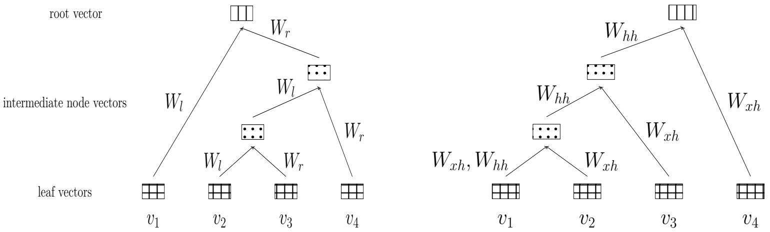
RctNN



LSTM


c_{t-2}
h_{t-2}
c_{t-1}
h_{t-1}
c_t
h_t
c_{t+1}
h_{t+1}
x_{t-1}
x_t
x_{t+1}
c_{t-1}
$\odot$
$+$
c_t
tanh
$\odot$
f_t
i_t
$\tilde{c}_t$
o_t
$\odot$
σ
σ
tanh
σ
h_{t-1}
cat
h_t
x_t

RNN vs RctNN

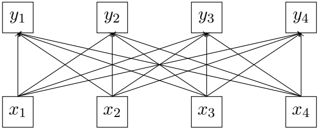


RNN is a generalized version of RctNN with a given tree structure.

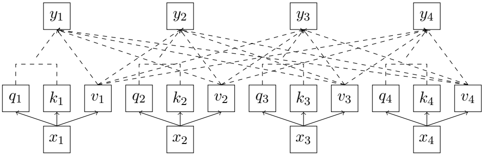
Transformers

- Multi-head self-attention (Vaswani, 2017).
- Transformer encoder (self-attention + Position-wise Feed-Forward Networks which is basically two linear transformations with a ReLU activation in between.)

The self-attention can be viewed as a dynamic fully-connected layer which does not need to change its weight matrix shape for different sized inputs.



(a) Fully-connected layer connection



(b) Self-attention layer connection

The solid lines represent a linear transformation with learnable parameters and the dashed lines mean a dot-product which does not require any parameters.

Pretrained transformers

- Most pre trained language models with transformers **do not** modify the basic transformer cell architecture. What they propose is on the training method of the networks.
- We tested BERT-multilang for cross-lingual tasks. Recently we have also qualitatively tested XLM and CamemBert in auto-complete scenario.

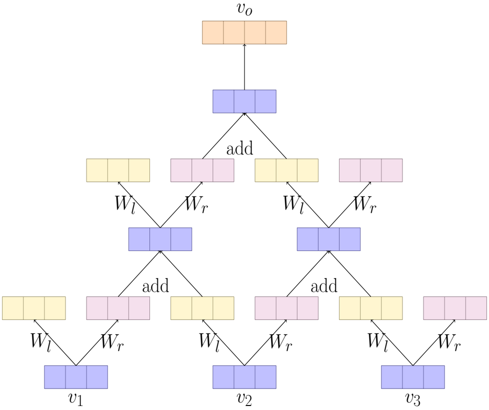
From what we have seen...

- Traditional NNs such as CNN and LSTM have obtained sota results on sequence tasks. In our case they are worse than the addition baseline.
- Using pre-trained language models (BERT) with fine tuning has set new standards in NLU (glue) tasks. However, apart from the fact that the tasks are sentence-level, fine-tuning such large model with modest size dataset seems to be less effective.
- Using pre-trained models with feature extraction has obtained very similar results. We would like to incorporate the pre-trained language models with feature extraction as a substitution of word2vec.
- Recursive neural network seems promising as it captures the sequence inner relation and fits more short sequences.

What we propose...

- Similar to RNN that captures the word inner relation.
- Similar to RNN that encodes a sequence into one vector without pooling operation.
- Does not need information other than the text. (tree-free)

TF-RNN



TF-RNN

$$
\begin{aligned}
v_{i,l}^{j} &= \tanh(W_l v_i^j + b_l) \\
v_{i-1,r}^{j} &= \tanh(W_r v_{i-1}^j + b_r) \\
v_i^{j+1} &= \tanh(v_{i,l}^j + v_{i-1,r}^j) \\
&... \\
v_o &= \tanh(U v_0^n + b)
\end{aligned} \tag{1}
$$

- $W_l \in \mathbb{R}^{d*d}$ et $W_r \in \mathbb{R}^{d*d}$ the weight matrices for associating the left and right context.
- A vector of the layer $j+1$ is calculated acccording to a pair of adjacent nodes from the previous layer j.
- $U \in \mathbb{R}^{p*d}$. Each linear transformation is followed by an activation.

Complexity

	RctNN	CNN	Self-Att	TF-RNN
Computational complexity	$O(n \cdot d^2)$	$O(k \cdot n \cdot d^2)$	$O(n^2 \cdot d)$	$O(\frac{n(n-1)}{2} \cdot d^2)$
Dependency length	$O(n)$	$O(\frac{n}{k})$	$O(1)$	$O(\frac{n}{2})$

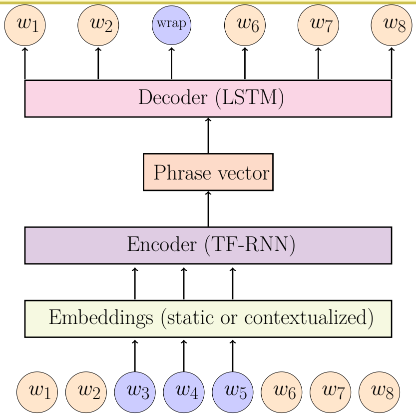
n = sequence length, d = model dimension where we assume $d = d_{input} = d_{hidden} = d_{output}$ for simplifying the comparison. k = kernel width for CNN.

- Quadratic with regard to n. But we treat only the short sequences, n is very small.
- The complexity for the self attention is only one head in one layer, so in practice it should be multiplied by $n_h * l$ where n_h is the head number and l the layer number.

How to train the network?

- Encoder-decoder
  - We have proposed a new training objective: "Wrapped context prediction"
  - Predict the context around a phrase with a specific vector to replace the phrase so that the generated sequence is syntactically complete.
    Normalization to sea level air density has no significant effect on the result . $\rightarrow$ *Normalization to sea level [PHR] has no significant effect on the result . [EOS]*
  - Can be seen as a conditional Skip-gram. In place of using independent one hot vectors, it predicts a continuous sequence where each predicted token is related from an encoded representation.

Overview



- **Inner relation** is captured by the TF-RNN.
- **Context** information is captured by the training method.

Experiments

- Phrase synonymy on specialized domain.
- Phrase similarity on general domain.
- Three inputs: static, contextualized and extended skip-gram. The static vectors are domain-specific information reinforced. (Liu, 2018)
- Four encoders: CNN, RctNN, Transformer and TF-RNN.

Main results

	Method	Synonymy (MAP)			Similarity (Correlation)	
		WE-fr	WE-en	BC-en	SemEval13†	SemEval17
Baselines	Skip-gram-ext	<0.5	<0.5	23.30	0.378	**76.827**
	Static-mean	5.29	12.19	39.65	26.910	38.843
	BERT-reduce	4.07	10.44	26.04	0.754	12.735
	BERT-mean	4.49	16.59	36.58	19.482	36.378
	ELMo-reduce	1.54	4.09	26.23	35.112	37.968
	ELMo-mean	7.37	5.20	29.27	37.991	36.207
	ELMo-concat	8.97	9.60	28.28	36.233	31.420
Context based	Static-CNN (0.4M)	7.42	15.71	35.75	*(29.890)*	42.245
	Static-Recurrent (0.5M)	12.89	20.53	42.60	*(21.720)*	42.961
	Static-Transf. (5M)	4.62	15.82	35.90	***(39.524)***	49.324
	Static-TF-RNN (0.5M)	**15.06**	**33.47**	**44.84**	*(22.003)*	44.382

- Encoder-decoder systems have better results (on SemEval2013 we do not have textual corpus to train the network).
- TF-RNN has the best results on specialized domain.

Reinforced embeddings

Task	**Embeddings**		
	ELMo	BERT	Static
WE-fr	9.57	6.47	**15.06**
WE-en	21.39	26.66	**33.47**
BC-en	23.61	26.01	**44.84**
Semeval2013 †	**24.279**	3.262	22.003
Semeval2017	**47.703**	29.078	44.382

- Domain specific information is more important than more advanced architectures.
- ELMo is more efficient on general domain while BERT may need more data (831 phrases in SemEval2017) to be effective because it tokenizes words in *subwords*.

Wrapped context prediction

Task	Training objectives		
	Plain	Context	Wrapped
WE-fr	9.40	13.35	**15.06**
WE-en	30.08	32.85	**33.47**
BC-en	39.48	41.49	**44.84**
Semeval2013†	16.759	21.376	**22.003**
Semeval2017	39.223	43.079	**44.382**

- The context prediction is meaningful.
- Adding a universal token for phrases facilitates the learning.

Phrase analogy

TF-RNN	**ADDITION**
wind power capacity : hydrogen storage :: wind power tank : ?	
hydrogen storage device	storage of hydrogen
gas storage tank	hydrogen storage system
hydrogen storage system	**hydrogen storage device**
marine mammal : marine epifauna :: marine animal : ?	
marine fish	epifauna
marine environment	marine
marine site	non marine
safety equipment : safety standard :: safety glass : ?	
safety concept	glass fibre
safety requirement	make of glass
carbon glass	carbon glass

Top 3 phrases similar to $b - a + a'$

Word alignment

- **Mikolov, 2013** Mapping by a linear transformation (Bilingual Word Embedding).

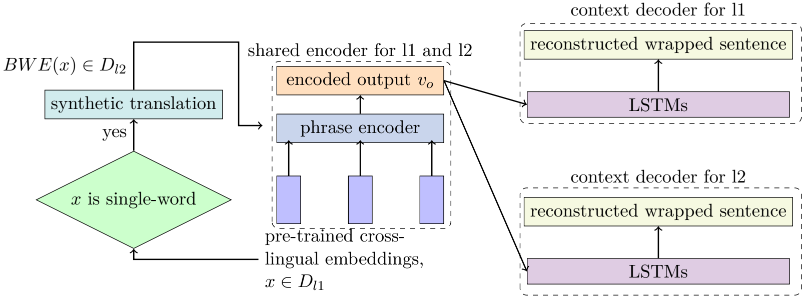
$$\arg\min_W \sum_i ||X_i W - Z_i||^2 \qquad (2)$$

- **Artexte et al 2018** A combination of several improvements.
  - whitening, orthogonal mapping, reweighting, dewhitening, dimension reduction
- **Lample and Conneau 2019** Pre-trained cross-lingual BERT. (XLM) (word or sequence level)
  - BERT + Translation Language Modeling (TLM)

Components

- **Encoder-decoder** has proven to be the best framework in our monolingual experiments.
- **Shared encoder**. We only use the encoder to generate bilingual multi-word representations once the training is completed.
- Training objective: **wrapped context prediction**. We can consider it as a *Denoising* objective in NMT.
- Pseudo **back-translation**: we use the bilingual word embedding approach of Artetxe, 2018 to generate synthetic translations in order to have pseudo parallel data for the single-words.

How to train?



$$\mathcal{L}_{cp\ l1\to l1}(\theta_{enc}, \theta_{dec_{\to l1}}) = -\mathbb{E}_{x\in D_{l1}} H(ws(x), dec_{\to l1}(enc(x))),$$

$$\mathcal{L}_{cp\ l2\to l1}(\theta_{enc}, \theta_{dec_{\to l1}}) = -\mathbb{E}_{x\in D_{l1}} H(ws(x), dec_{\to l1}(enc(\text{BWE}(x)))),$$

$$\mathcal{L}_{cp\ l2\to l2}(\theta_{enc}, \theta_{dec_{\to l2}}) = -\mathbb{E}_{x\in D_{l2}} H(ws(x), dec_{\to l2}(enc(x))),$$

$$\mathcal{L}_{cp\ l1\to l2}(\theta_{enc}, \theta_{dec_{\to l2}}) = -\mathbb{E}_{x\in D_{l2}} H(ws(x), dec_{\to l2}(enc(\text{BWE}(x))))$$

Experiments

- Bilingual phrase alignment.
- Two corpora in specialized domains. (WE, BC)
- Three language pairs. (en-es/fr/zh)
- Input: pre-trained reinforced cross-lingual fastText embeddings.
- Two baseline approaches. (Compositional method with context projection, Addition)
- Five different phrase encoders. (RctNN, CNN, LSTM, TXM, TF-RNN)
- Results in MAP. (mean average precision)

Main results

Dataset		Method		Encoder				Our
Corpus	Phrases	CMCBP	ADD	Rec.	CNN	LSTM	TXM	method
BC en-es	sw (72)	35.72	47.46	46.71	45.12	46.25	43.37	**47.76**
	n2n (21)	68.73	81.10	28.52	62.10	50.05	59.26	**86.11**
	p2q (9)	-	42.18	1.11	10.65	7.04	4.49	**49.11**
	all (108)	-	52.85	36.78	43.04	43.72	43.22	**55.40**
WE en-fr	sw (15)	65.56	78.25	77.22	78.33	79.36	**85.56**	79.44
	n2n (61)	42.09	57.37	6.16	40.84	18.64	41.82	**62.19**
	p2q (14)	-	15.83	<0.5	10.07	9.09	12.35	**37.95**
	all (90)	-	55.77	17.25	43.33	27.42	44.53	**62.10**
WE en-es	sw (15)	63.35	77.92	**88.89**	75.78	87.18	84.44	87.62
	n2n (61)	35.94	**62.68**	7.31	40.33	23.07	44.68	61.35
	p2q (14)	-	43.28	<0.5	28.57	17.86	37.20	**46.21**
	all (90)	-	62.20	19.77	44.41	32.94	50.14	**63.38**
WE en-zh	sw (17)	-	53.43	70.26	**76.47**	71.43	65.92	66.50
	n2n (47)	-	23.34	17.53	16.55	**25.24**	18.86	23.01
	p2q (26)	-	4.97	5.13	7.60	2.37	5.80	**12.32**
	all (90)	-	22.67	23.91	25.28	27.36	23.98	**28.13**
WE en-fr Liu2018	n2n (40)	67.32	78.36	46.07	68.51	44.82	48.47	**88.01**
	p2q (33)	-	34.38	2.38	20.01	7.93	28.25	**41.83**
	all (73)	-	58.48	26.06	46.59	28.13	39.33	**67.13**

- Our system has obtained the best overall results. Especially on the different length phrase alignment.
- The additive approach remains a solid approach. But between linguistically distant language pairs (English-Chinese), all encoder-decoder systems outperform the addition based approach.
- Transformer encoder is designed for long sequence tasks.

In single-word mode

	BC	WE		
Method	**en-es**	**en-fr**	**en-es**	**en-zh**
Mikolov, 2013	39.96	91.33	87.27	45.88
Artetxe, 2018	**49.13**	**95.56**	**90.39**	**73.52**
Our method	45.96	89.44	88.89	58.75

- Not too much degradation compared to Artetxe (2018).
- Better results compared to Mikolov et al. (2013).
- The proposal manages to hold a comparable performance for the standard bilingual word alignment task.

Examples

Dataset	Source	Addition	Our method
BC	breast cancer	cáncer mamario	cáncer de mama
en-es	cell death	muerte celular	muerte
WE	blade tip	angle des pales	côté supérieur de la pale
en-fr	Darrieus rotor	rotor tripale	rotor vertical
WE	airflow	freno aerodinámico	flujo de aire
en-es	wind power plant	electricidad del viento	planta eólica
WE	wind vane	偏航 ␣ 电机	风向标
en-zh	electricity power	电力	电力

Conclusion

- A new architecture to encode phrases in a unified way without the need for a syntax tree.
- A new training objective for unsupervised encoder-decoder systems.
- A pseudo *back-translation* to help the cross-lingual training.

Perspectives

- Extract-Edit (Wu, 2019) ⇒ synthetic translations for multi-words.

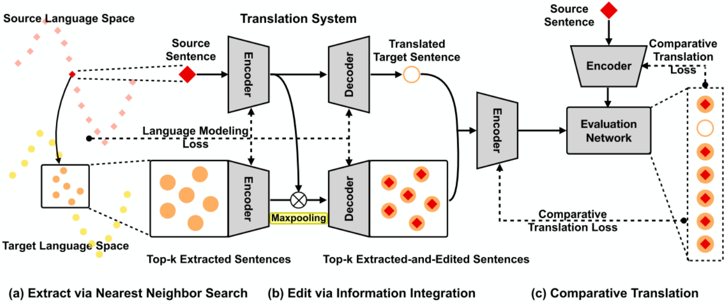


Diagram extracted from Wu et al, 2019

- XLM ⇒ end-to-end model but needs parallel data to fine-tune.

Bibliography

https://www.meetup.com/fr-FR/Nantes-Machine-Learning-Meetup/events/266311758/

Many thanks!