

Traduction automatique et attention

Introduction au TAL

Xiaoou Wang

Défis

- Faire comprendre l'utilité des réseaux de neurones dans la traduction automatique
- Contexte, développement et enjeux de la traduction automatique

Faire comprendre la langue aux machines

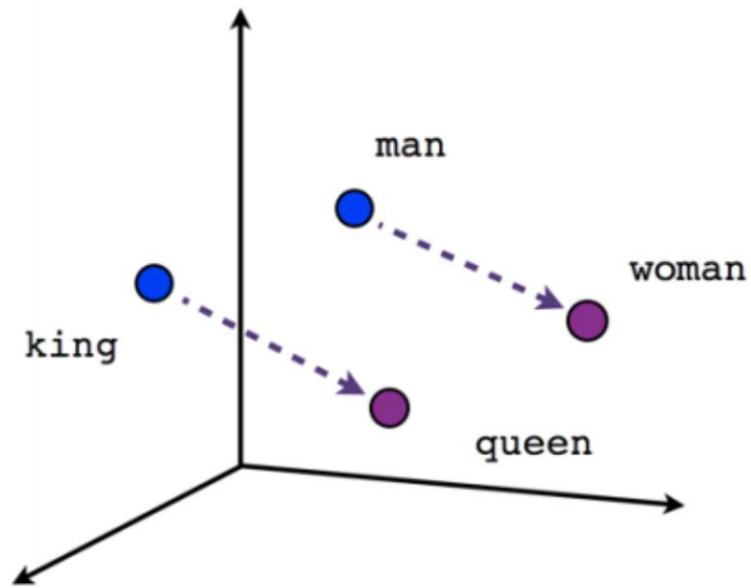
- La langue sous sa forme écrite est constituée d'unités discrètes

Pour l'ordinateur

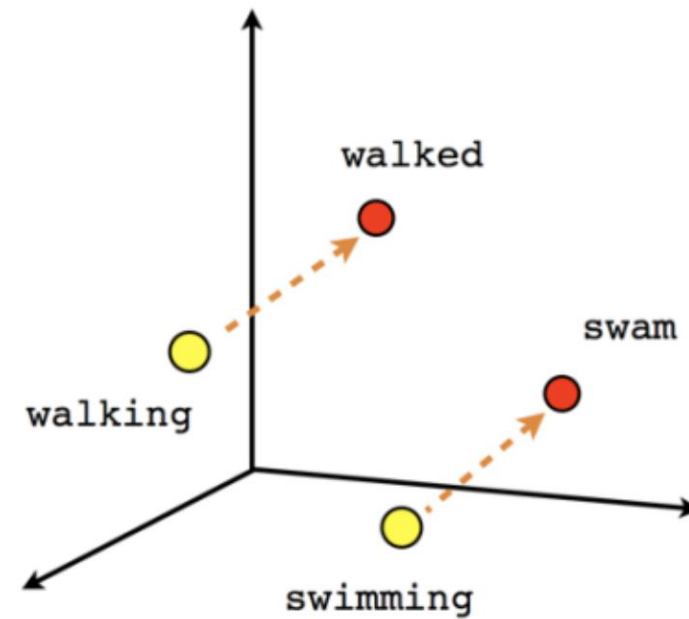
chat à char -> passage discret, il n'existe rien d'intermédiaire entre les deux unités

degré de noirceur pour les images (qch. entre chat et char)

Pour une représentation continue



Male-Female



Verb tense

Une représentation continue

- début : word2vec

hypothèse distributionnelle

Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov

Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen

Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado

Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean

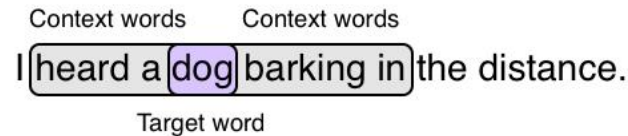
Google Inc., Mountain View, CA
jeff@google.com

**‘You shall know a word by
the company it keeps’**

Firth, John R., 1957. *Modes of meaning*. Oxford: Oxford University Press.

Implémentation

- Ce que l'on sait -> quels mots entourent le mot <dog> (distribution de probabilité)



- Supposons qu'on représente les mots <heard, a, dog, barking, in> par $[x_i, y_i]$, comment fait-t-on pour arriver à la distribution de $[0.1, 0.2, 0.3, 0.4]$?
- X_i, y_i -> word embeddings

Comment représenter une phrase ?

- Faire la moyenne des word embeddings

Jean adore le chat. = Le chat adore Jean.

Comment représenter une phrase ?

- RNN, un modèle autorégressif

Actualiser la représentation mentale d'une phrase au fil des mots.

Jean adore le chat.



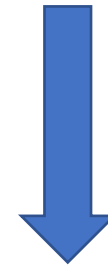
```
1 state = init_state()  
2 state = update(state, v("Jean"))  
3 state = update(state, v("adore"))  
4 state = update(state, v("le"))  
5 state = update(state, v("chat"))  
6 state = update(state, v("."))
```


Pourquoi représenter la phrase de manière continue

- Analyse sentimentale
- Classification de textes
- etc.

Améliorations

- On comprend mieux les mots quand on lit la phrase dans les deux sens.
- Jean lui montre le chat. (lui est souvent suivi d'un verbe et le précédé d'un verbe)

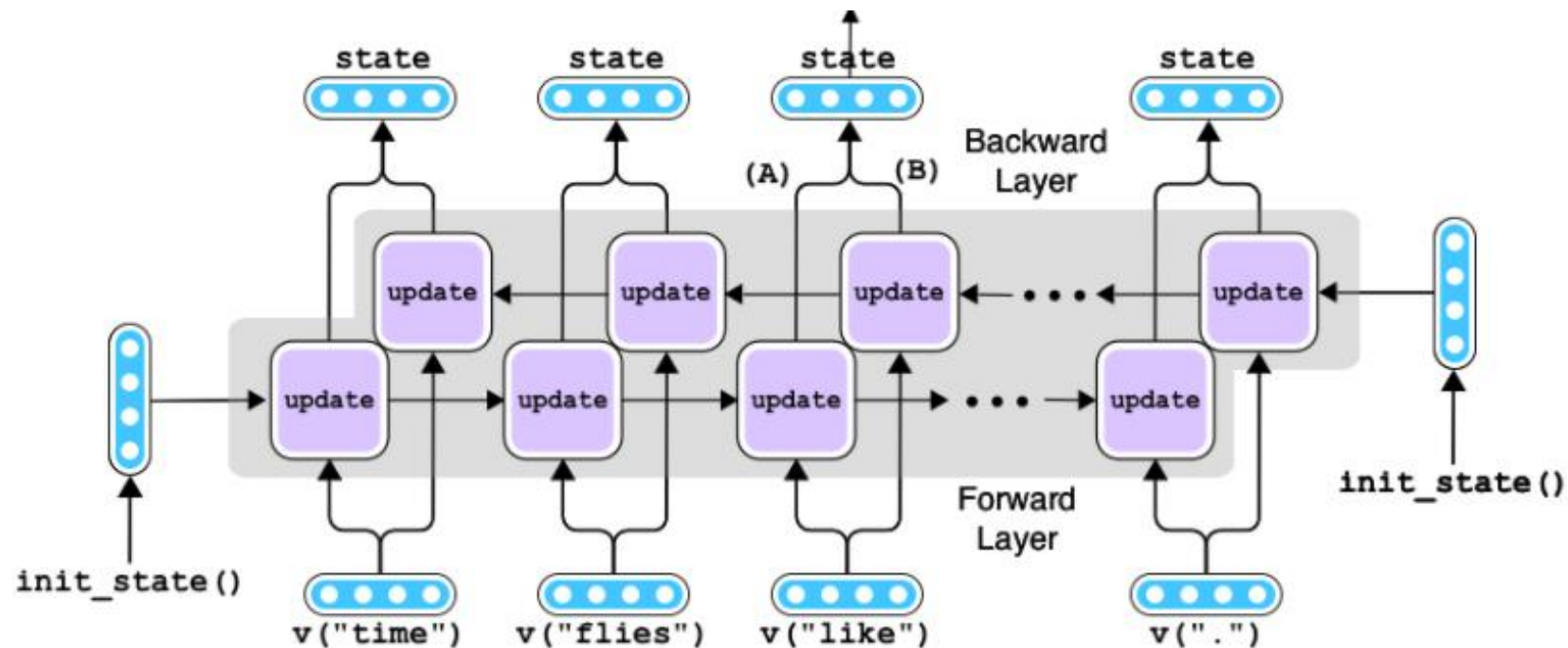


montre !=



RNN bidirectionnel

- mettre bout à bout deux embeddings (forward et backward)



Problème

- Plus la phrase est longue, plus il est difficile d'encoder la phrase.
- Principale contribution de l'article

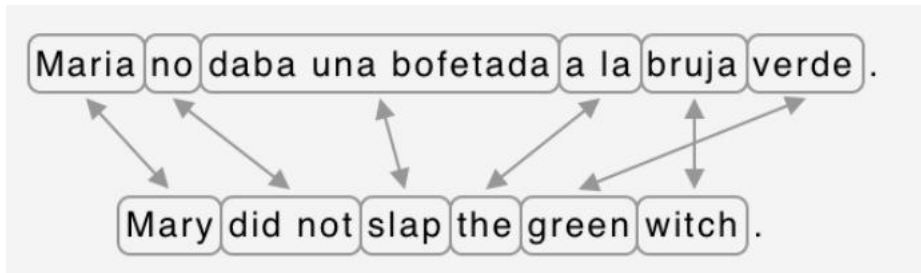
```
1 state = init_state()
2 state = update(state, v("Jean"))
3 state = update(state, v("adore"))
4 state = update(state, v("le"))
5 state = update(state, v("chat"))
6 state = update(state, v("."))
```

```
1 def rnn_simple(sentence):
2     word1, word2, word3, word4, word5, word6 = sentence
3     state = init_state()
4
5     state = f(w1 * f(w1 * f(w1 * f(w1 * f(w1 * f(w1 * state + w2 * word1 + b)
6         + w2 * word2 + b) + w2 * word3 + b) + w2 * word4 + b) + w2 * word5 + b) + w2
7         * word6 + b)
8     return state
```

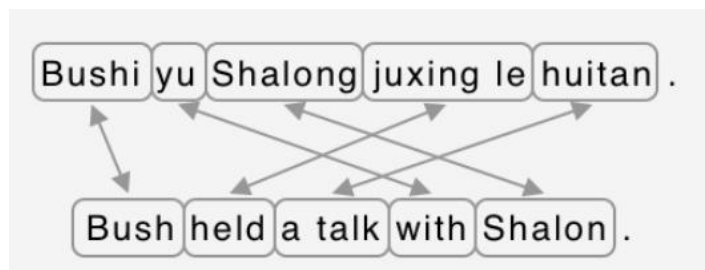
Petite histoire de machine translation

- système expert, Georgetown-IBM experiment, seconde guerre mondiale
- trouver la traduction d'un mot dans un dictionnaire de correspondance
- rédiger des règles pour aligner les mots
- Vite intenable !

spanish



chinese



japanese



1980s

- Statistical machine translation (SMT), plus de système expert
- utiliser un corpus bilingue (bitextes)
- trouver des traductions candidates susceptibles de correspondre au texte original

Moses




MOSES
statistical
machine translation
system

1. Moses

Overview
Manual 
Online Demos
FAQ
Mailing Lists
 Get Involved
Recent Changes

2. Getting Started

Source Installation
Baseline System
Packages
 Releases
Sample Data
Links to Corpora

3. Tutorials



[Main](#) » [HomePage](#)

Welcome to Moses!

Moses is a **statistical machine translation system** that allows you to automatically train translation models for any language pair. All you need is a collection of parallel corpora and a set of choices.

[Main](#) » [HomePage](#)

Welcome to Moses!

Moses is a **statistical machine translation system** that allows you to automatically train translation models for any language pair. All you need is a collection of parallel corpora and a set of choices.

News

- **5 October 2017** Moses v 4.0 has been **released!**
- **8 September 2016** [Moses2](#), a fast drop-in replacement for the Moses decoder
- **12 December 2015** [Add a new feature function to Moses](#)
- **17 June 2015** [Slate](#) for Windows
- **15 June 2015** Moses, and more, on Amazon cloud [Box](#)
- **1 June 2015** Developing Moses with Eclipse **video**
- **4 February 2015** Moses v 3.0 has been **released!**
- **21 July 2014** Moses now has nightly **speed tests**
- **14 July 2014** [How to compile Moses with Eclipse](#)
- **4 March 2014** Bug fix release for Moses, now version 2.1.1
- **3 February** The [2014 Machine Translation Marathon](#) will take place in Trento, Italy from 8-13th September.
- **21 January 2014** Moses v 2.1 has been **released!**
- **26 March 2013** The [2013 Machine Translation Marathon](#) (MTM2013) will take place in Prague, Czech Republic from 9-14th September
- **5 March 2013** What do you want to see in Moses v2.0? See [here](#) for projects and how to suggest them.
- **28 January 2013** Moses v 1.0 has been **released!**
- **12 October 2012** Moses v 0.91 released
- **February 2012:** Moses development is being supported by the EU under the **MosesCore** project
- **September 2011:** Moses now has a [cruise control page](#) to see the status of the current builds
- **September 2011:** Moses is now hosted on [github](#)

from 8-13th September.

e in Prague, Czech Republic from 9-14th September. Please email [mtm2013@prague.mt.ac.uk](#) to suggest them.

re project
ent builds

En 2015

- neural machine translation (NMT)

réseau de neurones (RNN bidirectionnel) + end-to-end

end-to-end, vous avez dit ?

Avantage de NMT

- «Unlike the traditional statistical machine translation, the neural machine translation aims at building a single neural network that can be jointly tuned to maximize the translation performance ``
 - Statistical machine translation (SMT), plus de système expert
 - utiliser un corpus bilingue (*bitextes*)
 - trouver des traductions candidates susceptibles de correspondre au texte original

De manière classique, SMT a besoin d'un modèle de langue pour choisir la bonne traduction, approche composée

Composante 1 -> trouver des bouts de traduction, composante 2 -> réagencement des segments 3 -> modèle de

Avantage de NMT

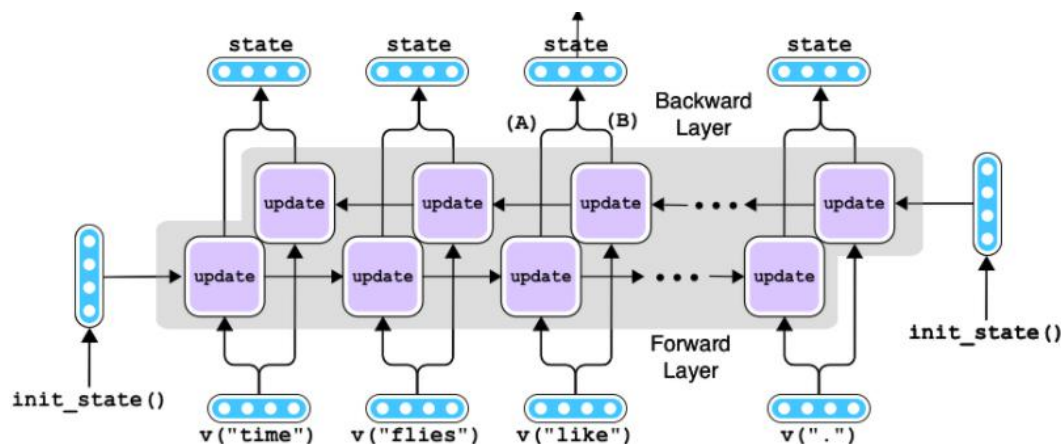
- En NMT, le système est structuré de manière à entraîner les composantes en même temps

Composante 1 (composante principale) : mécanisme d'attention

«We conjecture that the use of a fixed-length vector is a bottleneck in improving the performance of this basic encoder–decoder architecture, and propose to extend this by allowing a model to automatically (soft-)search for parts of a source sentence that are relevant to predicting a target word, without having to form these parts as a hard segment explicitly.»

Retournons au RNN

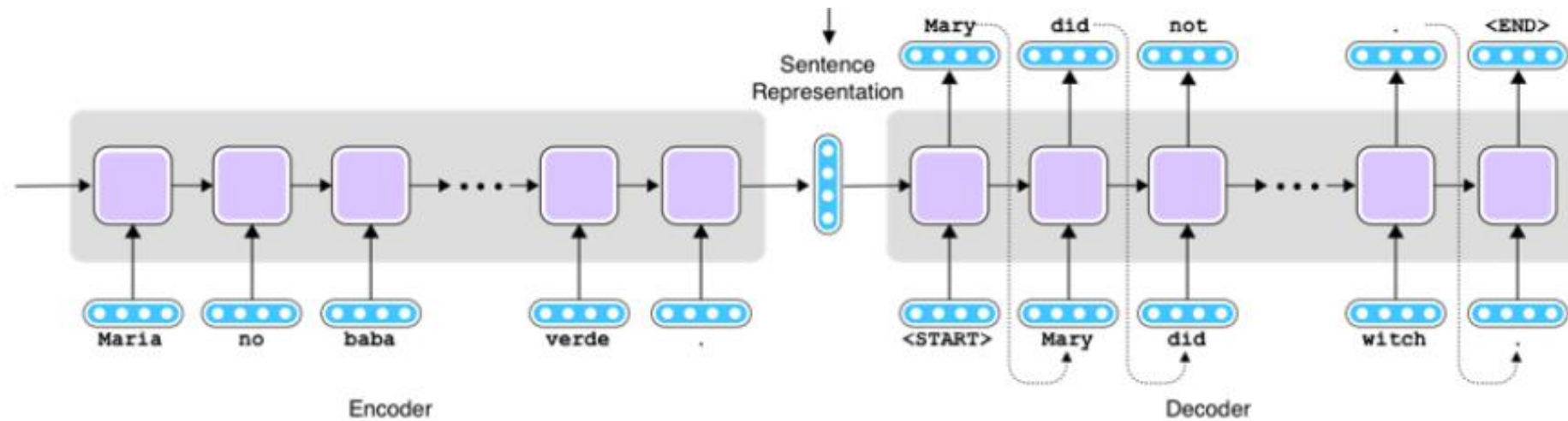
- Une phrase, quelle que soit sa longueur, est encodée dans un embedding [1.2,3.4,...,2.3]



The usual RNN, described in Eq. (1), reads an input sequence x in order starting from the first symbol x_1 to the last one x_{T_x} . However, in the proposed scheme, we would like the annotation of each word to summarize not only the preceding words, but also the following words. Hence, we propose to use a bidirectional RNN (BiRNN, Schuster and Paliwal, 1997), which has been successfully used recently in speech recognition (see, e.g., Graves *et al.*, 2013).

Retournons au RNN

Le décodeur (traducteur) ne dispose d'un seul embedding pour traduire toute la phrase, quelle que soit sa longueur.



Mécanisme d'attention

- Que fait un traducteur humain ?

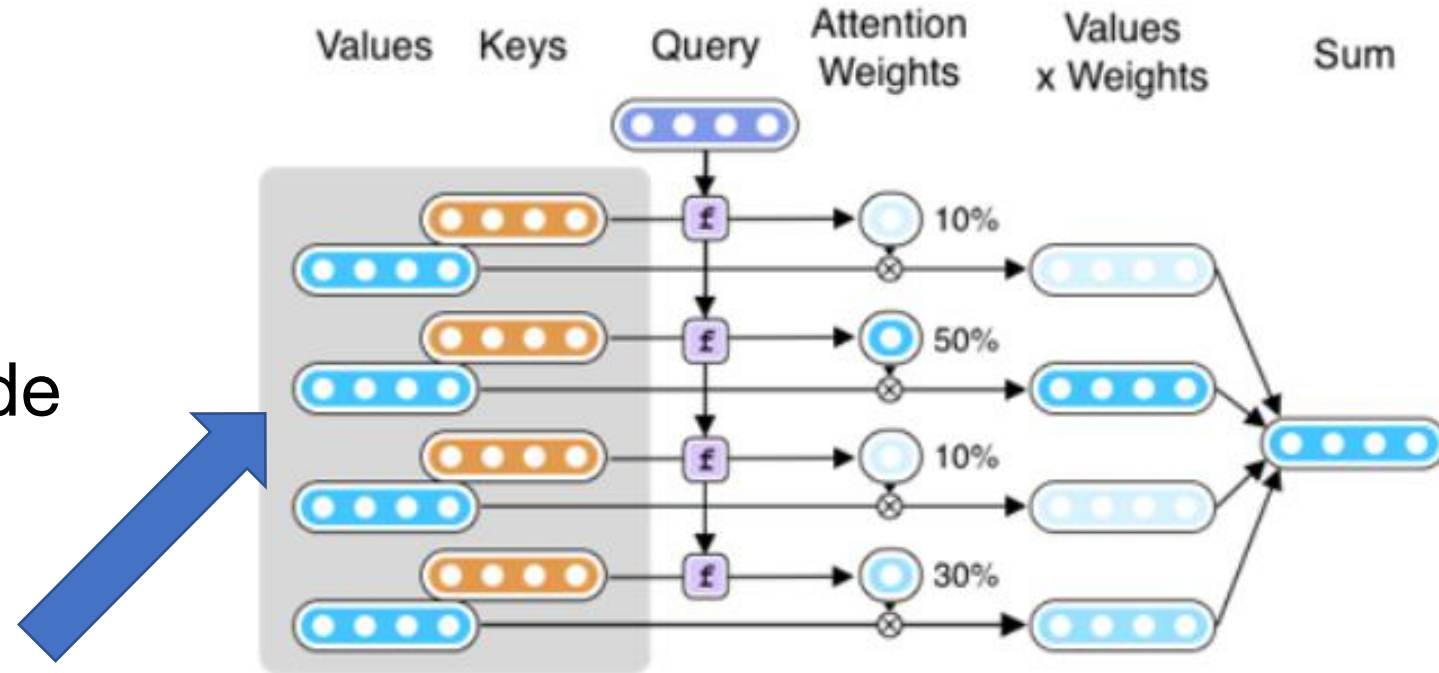
Un traducteur humain ne mémorise pas la phrase, il traduit petit à petit, en se référant constamment à la partie pertinente de la phrase d'origine.

Au fil de la traduction, il focalise son attention sur une partie de la phrase.

Implémenter l'attention

Éléments principaux :

1, keys = values =
les inputs (représentation de
chaque mot)

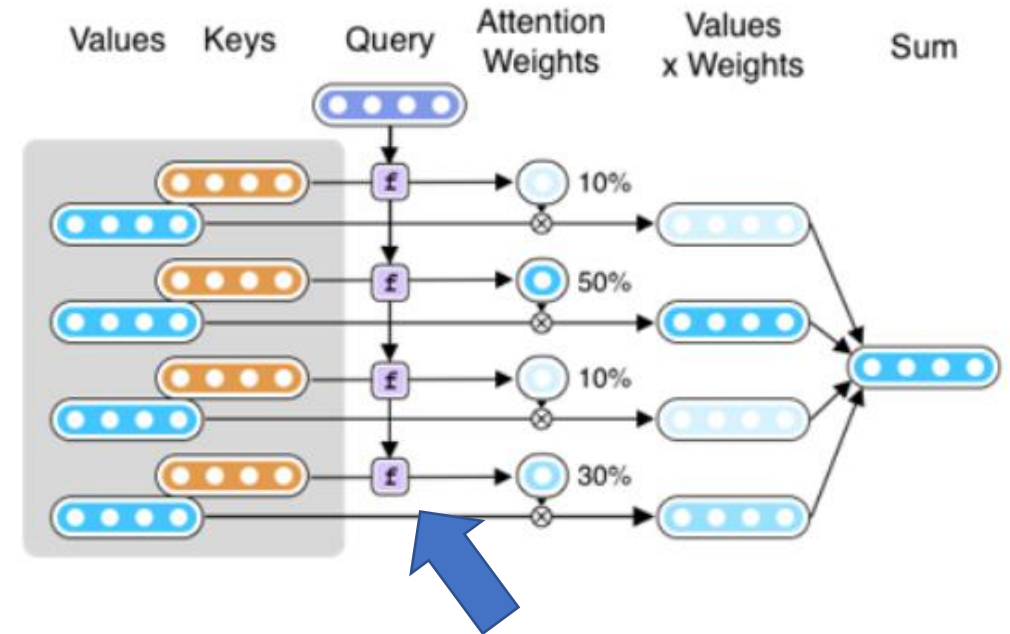


Implémenter l'attention

Éléments principaux :

1, keys = values =
les inputs (embeddings des mots)

2, Chaque key est comparée avec
une query (le mot à traduire) grâce à une fonction d'atten
f -----> une distribution de weights pour chaque
input. Plus le poids est élevé, plus cet input est pertinent

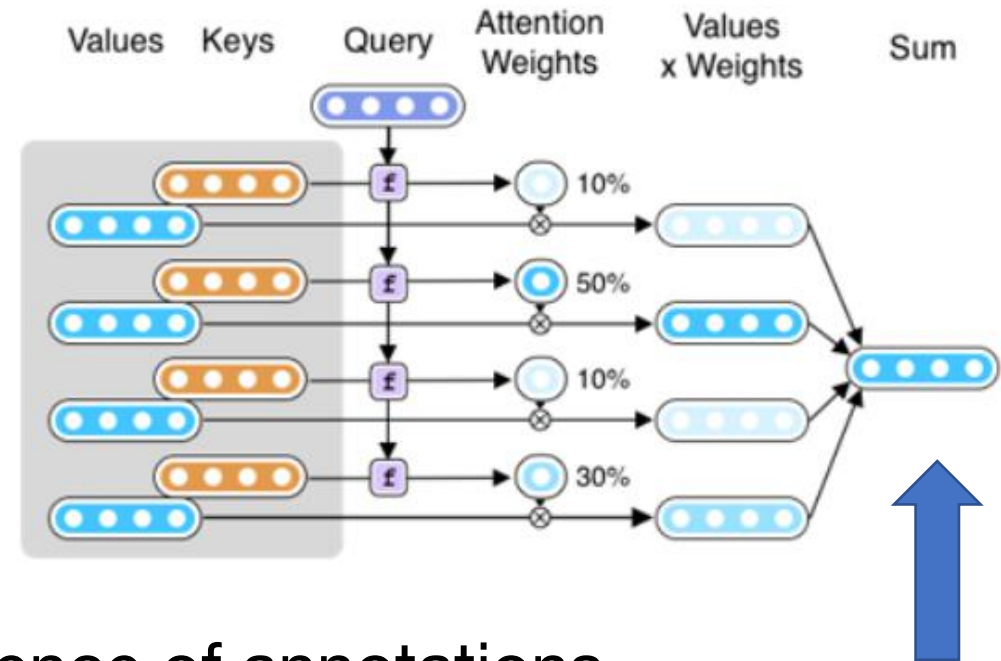


«The context vector C_i depends on a sequence of annotations (h_1, \dots, h_{T_x}) to which an encoder maps the input sentence. Each annotation h_i contains information about the whole input sequence with a strong focus on the parts surrounding the i -th word of the input sequence.»

Implémenter l'attention sur la machine

Éléments principaux :

3, Les inputs sont pondérés avec leur poids respectif pour produire une somme vectorielle.



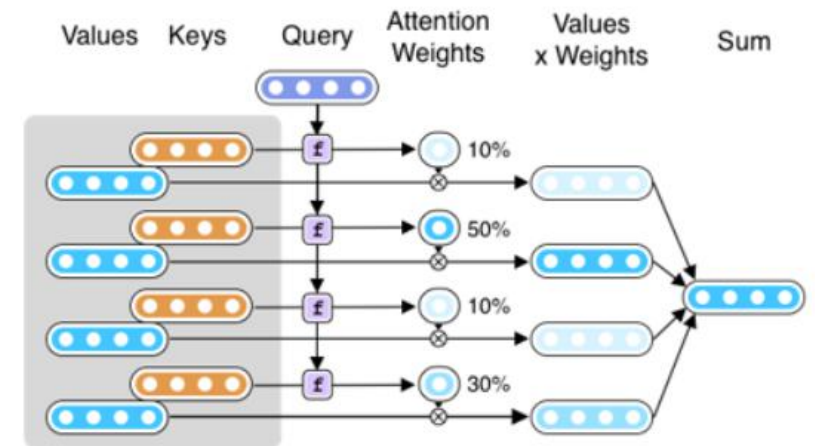
«The context vector C_i depends on a sequence of annotations (h_1, \dots, h_{T_x}) to which an encoder maps the input sentence. Each annotation h_i contains information about the whole input sequence with a strong focus on the parts surrounding the i -th word of the input sequence.»

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$$

Implémenter l'attention sur la machine

Éléments principaux :

Pour chaque nouveau mot à traduire (query)
le vecteur contextuel est donc distinct.



❖ Revisitons l'analogie avec le traducteur humain

Avantage de NMT

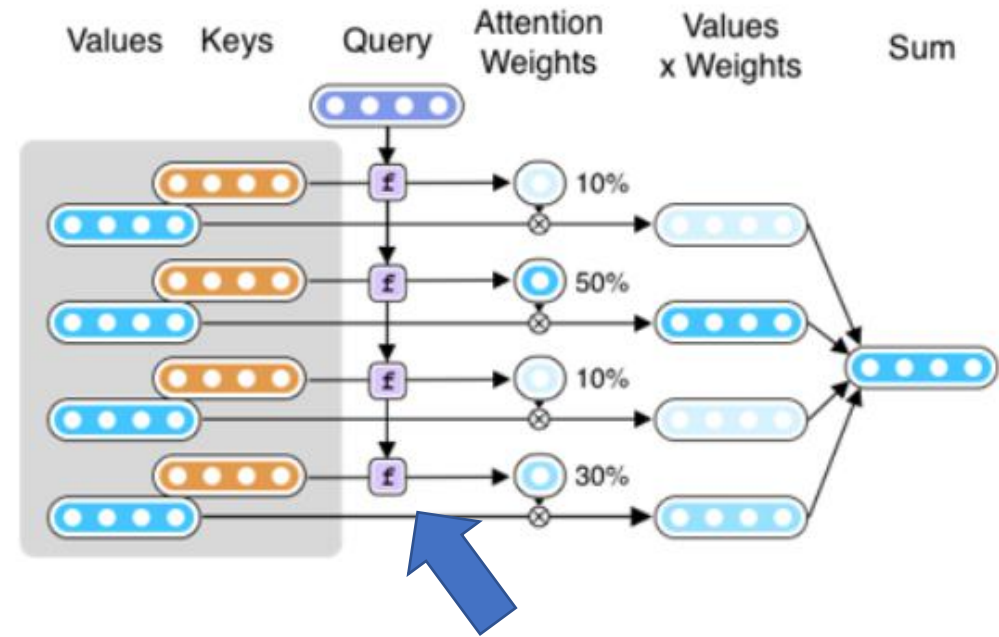
- En NMT, le système est structuré de manière à entraîner les composantes en même temps -> end-to-end

Autre bénéfice du vecteur contextuel : modèle d'alignement (composante 2)

We parametrize the alignment model a as a feedforward neural network which is jointly trained with all the other components of the proposed system.

We can understand the approach of taking a weighted sum of all the annotations as computing an expected annotation, where the expectation is over possible alignments.

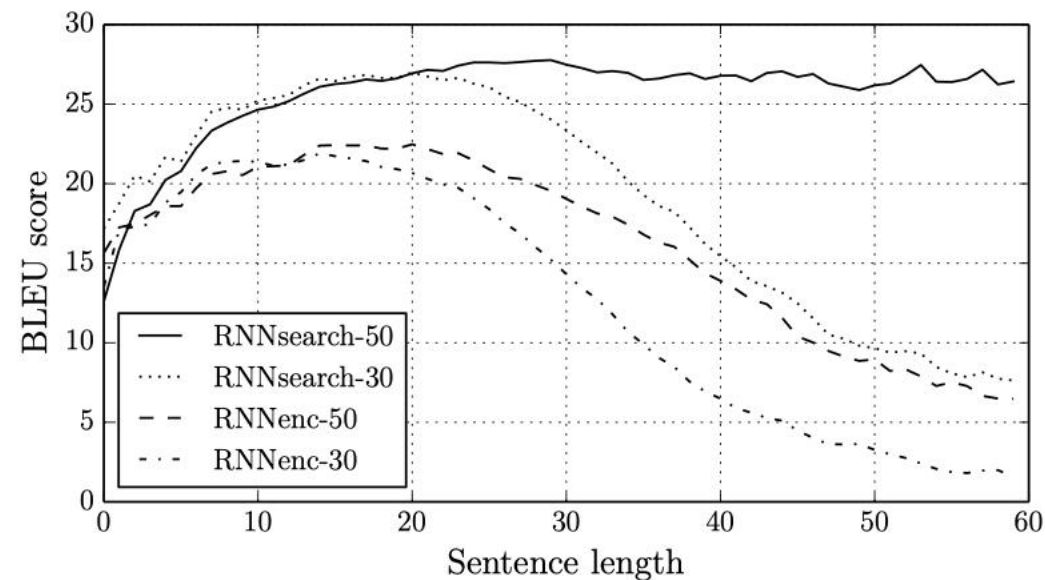
Composante 3



2, Chaque key est comparée avec une query (le mot à traduire) grâce à une fonction d'attention f -> une distribution de weights pour chaque input. Plus le poids est élevé, plus cet input est pertinent.

Résultat quantitatif

- RNNsearch = RNN avec attention, RNNenc = RNN sans attention
- WMT '14 (corpus parallèle français-anglais)
- corpus : 348M mots | test : 3003 phrases

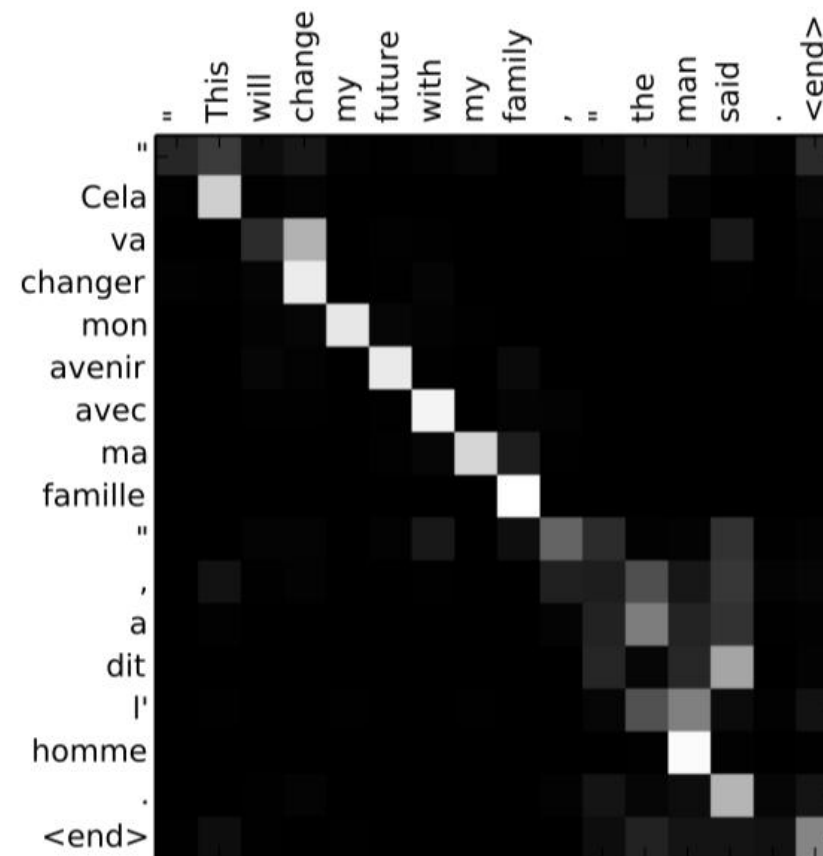


➤ «One of the motivations behind the proposed approach was the use of a fixed-length context vector in the basic encoder-decoder approach. We conjectured that this limitation may make the basic encoder-decoder approach to underperform with long sentences.»

- RNNsearch50, especially, shows no performance deterioration even with sentences of length 50 or more.
- RNNsearch-30 even outperforms RNNencdec-50.

Résultat d'alignement

- «The strength of the soft-alignment, opposed to a hard-alignment, is evident. Any hard alignment will map [the] to [l'] and [man] to [homme]. This is not helpful for translation
- An additional benefit of the soft alignment is that it naturally deals with source and target phrases of different lengths, without requiring a counter-intuitive way of mapping some words to or from nowhere ([NULL])



(d)

Conclusions et contributions principales

- Le réseau de neurones conçu par les auteurs forme un système end-to-end, alors que les travaux précédents utilisent une approche considérant le réseau de neurones comme une composante d'un système (souvent SMT) dont le rôle est restreint (choisir la meilleure traduction candidate)
- Les approches précédentes utilisent un seul embedding pour encoder les phrases, entraînant une chute de performance lorsque la longueur de phrase est supérieure à 30 tokens. Le mécanisme d'attention proposé par les auteurs de l'article a permis d'éviter cette chute de performance sur la traduction de l'anglais vers le français.
- Cet article constitue une véritable révolution et a inspiré le célèbre article « attention is all you need » de Vaswani, Ashish, et al.



Références principales

- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
- Goldberg, Yoav. "A Primer on Neural Network Models for Natural Language Processing." ArXiv:1510.00726 [Cs], October 2, 2015. <http://arxiv.org/abs/1510.00726>.
- Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep learning (Vol. 1, No. 2). Cambridge: MIT press.