Large scale, maximum margin regression based, structural learning approach to phrase translations

Sandor Szedmak[1]

joint work with

Esther Galbrun[43] Craig Saunders[2], Yizhao Ni[1]

[1]University of Southampton [2]XRCE XEROX [3]University of Helsinki [4]INSA of Rouen

EAMT 2009 Barcelona

Outline

Main Components of the translator system

- Phrase translator - the main topic of this presentation.
  - A well known system: GIZA++
  - Additional postprocessing tools, e.g. in Moses
- Decoder, which can fit better to the phrase dictionary generated by maximum margin learning procedure.

The base learning problem of phrase translation

- A phrase implies a binary classification of the words of a sentence;
  - the words within the phrase are the positive cases,
  - the remaining part gives the negative ones.
- The translation can be interpreted as a propagation of the classes of a source sentence into the corresponding target sentence.
- It might be interpreted either as an inductive or a transductive learning problem.

The learning schema

class	source words	predicted class	target words
–	*Je*	$?(+,-)$	*I*
–	*vous*	$?(+,-)$	*would*
–	*demands*	$?(+,-)$	*therefore*
–	*donc*	$?(+,-)$	*once*
–	*à*	$?(+,-)$	*more*
–	*nouveau*	$?(+,-)$	*ask*
–	*de*	$?(+,-)$	*you*
–	*faire*	$?(+,-)$	*to*
–	*le*	$?(+,-)$	*ensure*
–	*nécessaire*	$?(+,-)$	*that*
–	*pour*	$?(+,-)$	*we*
+	**que**	$?(+,-)$	*get*
+	**nous**	$?(+,-)$	*a*
+	**puissions**	$?(+,-)$	*Dutch*
+	**disposer**	$?(+,-)$	*channel*
–	*d'*	$?(+,-)$	*as*
–	*une*	$?(+,-)$	*well*
–	*chaîne*		
–	*néerlandaise*		

Computational difficulties

- If the sentence length in words is 30 and the maximum length allowed of non-gapped phrases is 5 then **140 binary classification problems have to be solved!**
- *Does any acceptable efficient joint approximation schema exist at all?*

A learning approach

- The **Support Vector Machine(SVM)** has proved to be a highly accurate learning tool, but it is able to deal only with binary outputs.
- The learning framework of the **SVM** can be extended to predict arbitrary vector represented outputs with no additional cost, we will call it **Maximum Margin Regression(MMR)** in the sequel.
  - The details are discussed when the concrete learning problem is unfolded.
  - MATLAB source code of the solver and a demo for multiclass classification in MMR is freely available on the web.

The skeleton of the phrase translation

Sentence-wise word relations, the building blocks:

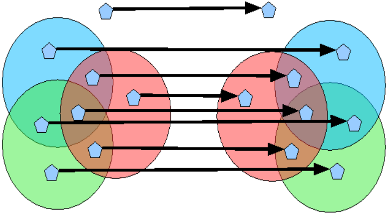
- global relationships between word pairs,
- local relations,
- inference between global and local relations,

Estimating phrases,
- Collect those sequences of source and target words which have the highest accumulated word-wise relations.

A projection rule of the sentences

Mapping words

Mapping phrases

- 

$$\begin{gathered}\mathcal{P}_1 \Leftrightarrow \mathcal{R}_1, \\ \mathcal{P}_2 \Leftrightarrow \mathcal{R}_2, \\ \mathcal{P}_1 \cap \mathcal{P}_2 \Leftrightarrow \mathcal{R}_1 \cap \mathcal{R}_2, \\ \mathcal{P}_1 \cup \mathcal{P}_2 \Leftrightarrow \mathcal{R}_1 \cup \mathcal{R}_2, \\ \mathcal{P}_1 \setminus \mathcal{P}_2 \Leftrightarrow \mathcal{R}_1 \setminus \mathcal{R}_2, \\ \mathcal{P}_2 \setminus \mathcal{P}_1 \Leftrightarrow \mathcal{R}_2 \setminus \mathcal{R}_1.\end{gathered}$$

- Intersections mapped into corresponding intersections of the subsets of words those we might consider as phrases. Obviously it can be achieved only approximately!

Global versus local relations of words

- Interference of global and local relations:
  - Strong global: Frequent co-occurrences,
  - Strong local: adjacent(or almost adjacent) positions
	Globally weak	Globally strong
Locally weak	High confidence No relation	Likely No relation
Locally strong	Likely $\star$ There is a relation	High confidence There is a relation
- $\star$ case of rare words!

Sentence-wised word distances

Distances:

- The distances measure the co-occurrences of words and their relative positions within the sentences.
  - A co-occurrence with high distance is down scaled.
- Within a language:

$$d_{\mathcal{S}}(w_1, w_2) = \min_{i_1 \in I(w_1), i_2 \in I(w_2)} \left| \frac{i_1}{n_S} - \frac{i_2}{n_S} \right|$$

- Between two languages:

$$d_{\mathcal{S}_s,\mathcal{S}_t}(w_1, w_2) = \min_{i_1 \in I(w_1), i_2 \in I(w_2)} \left| \frac{i_1}{n_{S_s}} - \frac{i_2}{n_{S_t}} \right|$$

Sentence-wised word similarities

Similarities:

- Linear:

$$s_{\mathcal{S}}(w_1, w_2) = 1 - d_{\mathcal{S}}(w_1, w_2)$$
$$s_{\mathcal{S}_s,\mathcal{S}_t}(w_1, w_2) = 1 - d_{\mathcal{S}_s,\mathcal{S}_t}(w_1, w_2)$$

- Gaussian:

$$s_{\mathcal{S}}(w_1, w_2) = e^{\left(-\frac{d^2_{\mathcal{S}}(w_1,w_2)}{\sigma}\right)}$$
$$s_{\mathcal{S}_s,\mathcal{S}_t}(w_1, w_2) = e^{\left(-\frac{d^2_{\mathcal{S}_s,\mathcal{S}_t}(w_1,w_2)}{\sigma}\right)}$$

- Logistic:

$$s_{\mathcal{S}}(w_1, w_2) = \frac{1}{4s}\operatorname{sech}^2\left(\frac{d_{\mathcal{S}}(w_1,w_2)}{2s}\right)$$
$$s_{\mathcal{S}_s,\mathcal{S}_t}(w_1, w_2) = \frac{1}{4s}\operatorname{sech}^2\left(\frac{d_{\mathcal{S}_s,\mathcal{S}_t}(w_1,w_2)}{2s}\right)$$
$$\operatorname{sech}(z) = \frac{1}{\cosh(z)} = \frac{2}{e^z+e^{-z}}$$

Global(training set relative) similarity

- Within a language:

$$s(w_1, w_2) = \frac{\sum_{S \in \mathcal{S}(w_1) \cap \mathcal{S}(w_2)} s_{\mathcal{S}}(w_1, w_2)}{|\mathcal{S}(w_1) \cup \mathcal{S}(w_2)|}$$

- Between two languages:

$$s(w_1, w_2) = \frac{\sum_{S \in \mathcal{S}_s(w_1) \cap \mathcal{S}_t(w_2)} s_{\mathcal{S}_s, \mathcal{S}_t}(w_1, w_2)}{|\mathcal{S}(w_1)_s \cup \mathcal{S}_t(w_2)|}$$

$\mathcal{S}(w)$ is the index set of the sentences containing word w in the training set.

Word features, local relations

Word features with respect to a sentence pair(source-target) expressed as a concatenated vector of the similarities between the word and the words of the source and the target sentences.

- Source words:

$$\phi_{\mathcal{S}_s,\mathcal{S}_t}(w_s) = \overbrace{\underbrace{(s(w_s, w_{s_1}), \ldots, s(w_s, w_{s_{n_{\mathcal{S}_s}}}}_{(w_{s_1},\ldots,w_{s_{n_{\mathcal{S}_s}}})=\mathcal{S}_s}}^{\text{Relations to the source}}), \overbrace{\underbrace{s(w_s, w_{t_1}), \ldots, s(w_s, w_{t_{n_{\mathcal{S}_t}}})}_{(w_{t_1},\ldots,w_{t_{n_{\mathcal{S}_t}}})=\mathcal{S}_t}}^{\text{Relations to the target}}$$

- Target words:

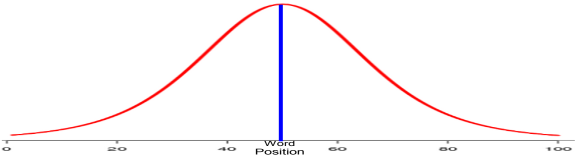
$$\phi_{\mathcal{S}_s,\mathcal{S}_t}(w_t) = \overbrace{\underbrace{(s(w_t, w_{s_1}), \ldots, s(w_t, w_{s_{n_{\mathcal{S}_s}}}}_{(w_{s_1},\ldots,w_{s_{n_{\mathcal{S}_s}}})=\mathcal{S}_s}}^{\text{Relations to the source}}), \overbrace{\underbrace{s(w_t, w_{t_1}), \ldots, s(w_t, w_{t_{n_{\mathcal{S}_t}}})}_{(w_{t_1},\ldots,w_{t_{n_{\mathcal{S}_t}}})=\mathcal{S}_t}}^{\text{Relations to the target}})$$

Feature = Language model + Translation model

$$\phi_{\mathcal{S}_s,\mathcal{S}_t}(\mathcal{S}_s,\mathcal{S}_t) = \left[\begin{array}{cc} SS & ST \\ TS & TT \end{array}\right],$$

- SS relationship between source items,
- TT relationship between target items,
- ST(TS) relationship between source and target items.

Word positions

- The position feature of a word should express the uncertainty arising from the varying grammatical relations.
- This uncertainty can be captured by a probability density function with an expected value localized in the real position of the word in a given concrete sentence.
- $\psi_S(w) = f(.|i_w, \Theta)$, where
  - f a suitable density function, e.g. Gaussian
  - i_w is the position of the word in sentence $\mathcal{S}$,
  - Θ a scale parameter, e.g. variance,
- 

Learning problem

- The densities are the representation of the assumed to be correct positions are inferred with features as representation of the relations of the words.
- We predict:

word relations

$\Updownarrow$

expected position of the words within a sentence

Optimization problem

- Optimization framework, Maximum Margin Regression(MMR):

$$
\begin{array}{ll}
\min & \frac{1}{2}\|\mathbf{W}\|^2_{Frobenius} + C\sum_{s=1}^{n_{\mathcal{S}_s}} \xi_s \\
\text{w.r.t.} & \mathbf{W} \text{ linear operator}, \boldsymbol{\xi} \text{ loss}, \\
\text{s.t.} & \langle \underbrace{\psi_{\mathcal{S}_s}(w_s)}_{\text{possible word position}}, \mathbf{W}\underbrace{\phi_{\mathcal{S}_s,\mathcal{S}_t}(w_s)}_{\text{word relations}}\rangle \geq 1 - \xi_s,\ w_s \in \mathcal{S}_s, \\
 & \boldsymbol{\xi} \geq \mathbf{0},
\end{array}
$$

- The optimum has the form:

$$\mathbf{W} = \sum_{w_s \in \mathcal{S}_s} \alpha_{w_s} \psi_{\mathcal{S}_s}(w_s) \phi_{\mathcal{S}_s,\mathcal{S}_t}(w_s)',$$

High level, margin based word similarity measure

- Sentence relative similarity predicted between all pairs of source and target words:

$$
\begin{array}{ll}
\multicolumn{2}{c}{\text{source} \Rightarrow \text{target}} \\
\hline
\mathscr{R}_{\mathbf{W}}(w_s, w_t) & = \langle \psi_{\mathcal{S}_s}(w_s), \mathbf{W}\phi_{\mathcal{S}_s,\mathcal{S}_t}(w_t) \rangle \\
& = \sum_{w_r \in \mathcal{S}_s} \alpha_{w_r} \kappa_\psi(w_s, w_r) \kappa_\phi(w_r, w_t)
\end{array}
$$

and

$$
\begin{array}{ll}
\multicolumn{2}{c}{\text{target} \Rightarrow \text{source}} \\
\hline
\mathscr{R}'_{\mathbf{W}}(w_t, w_s) & = \langle \mathbf{W}'\psi_{\mathcal{S}_s}(w_s), \phi_{\mathcal{S}_s,\mathcal{S}_t}(w_t) \rangle \\
& = \sum_{w_r \in \mathcal{S}_s} \alpha_{w_r} \kappa_\psi(w_s, w_r) \kappa_\phi(w_r, w_t)
\end{array}
$$

where

$$
\kappa_\psi(w_s, w_r) = \langle \psi_{\mathcal{S}_s}(w_s), \psi_{\mathcal{S}_s}(w_r) \rangle
$$
$$
\kappa_\phi(w_r, w_t) = \langle \phi_{\mathcal{S}_s,\mathcal{S}_t}(w_r), \phi_{\mathcal{S}_s,\mathcal{S}_t}(w_t) \rangle.
$$

Word alignment

- A *source word* is aligned to those *target words* which **maximizes the relations**, and
a *target word* is aligned to *those source words* which **maximizes the relations**

$$
\begin{array}{ll}
w_s \Leftrightarrow w_t & \hat{w}_s(w_t) \in \arg\max_{w \in \mathcal{S}_s} \mathscr{R}_{\mathbf{W}}(w, w_t), \\
w_t \Leftrightarrow w_s & \hat{w}_s(w_s) \in \arg\max_{w \in \mathcal{S}_t} \mathscr{R}'_{\mathbf{W}}(w, w_s).
\end{array}
$$

- The words can be aligned to more than one words in ambiguous cases!

Alignment of four views

- **W** computed on the source words only and the target labels are predicted

$$\begin{array}{ll} w_s \Leftrightarrow w_t & \hat{w}_s(w_t) \in \arg\max_{w \in \mathcal{S}_s} \mathscr{R}_{\mathbf{W}_s}(w, w_t), \\ w_t \Leftrightarrow w_s & \hat{w}_s(w_s) \in \arg\max_{w \in \mathcal{S}_t} \mathscr{R}'_{\mathbf{W}_s}(w, w_s). \end{array}$$

- **W** computed on the target words only and the source labels are predicted

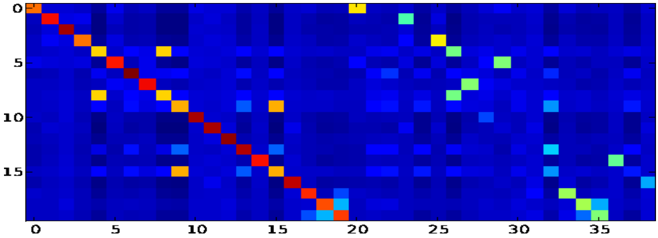
$$\begin{array}{ll} w_s \Leftrightarrow w_t & \hat{w}_s(w_t) \in \arg\max_{w \in \mathcal{S}_s} \mathscr{R}_{\mathbf{W}_t}(w, w_t), \\ w_t \Leftrightarrow w_s & \hat{w}_s(w_s) \in \arg\max_{w \in \mathcal{S}_t} \mathscr{R}'_{\mathbf{W}_t}(w, w_s). \end{array}$$

Example sentences

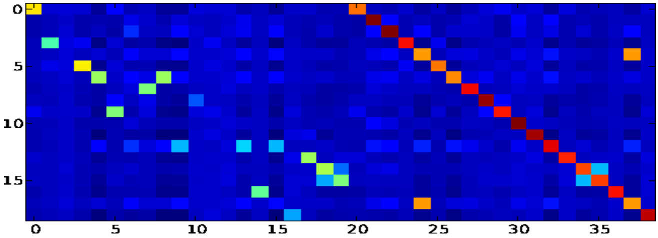
source words	word index	target words	word index
Je	0	I	0
vous	1	would	1
demands	2	therefore	2
donc	3	once	3
à	4	more	4
nouveau	5	ask	5
de	6	you	6
faire	7	to	7
le	8	ensure	8
nécessaire	9	that	9
pour	10	we	10
que	11	get	11
nous	12	a	12
puissions	13	Dutch	13
disposer	14	channel	14
d’	15	as	15
une	16	well	16
chaîne	17		
néerlandaise	18		

Features, as they look like

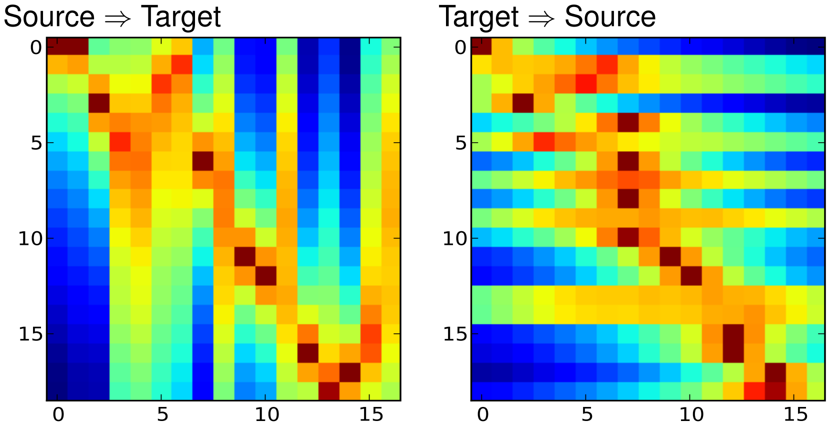
Feature values to the source



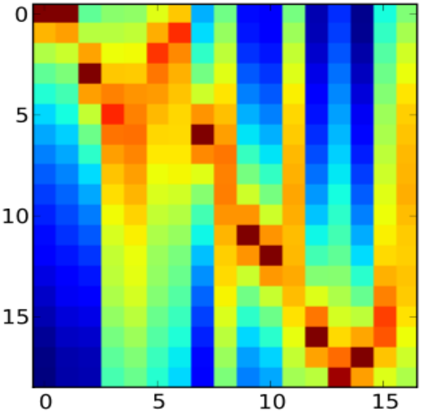
Feature values to the target



Word relations



Word relations



Je	0	I	0
vous	1	would	1
demands	2	therefore	2
donc	3	once	3
à	4	more	4
nouveau	5	ask	5
de	6	you	6
faire	7	to	7
le	8	ensure	8
nécessaire	9	that	9
pour	10	we	10
que	11	get	11
nous	12	a	12
puissions	13	Dutch	13
disposer	14	channel	14
d’	15	as	15
une	16	well	16
chaîne	17		
néerlandaise	18		

Alignment, four views

The relations between words can be reduced to the raw and column maximums (they might be not unique).
They can express edges between words in a word graph.

Training: source   source words ⇒ target words

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	6	5	2	3	3	7	8	8	8	8	9	10	10	15	15	12	14	13

Training: target   source words ⇒ target words

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	6	5	2	7	3	7	7	7	8	7	9	10	11	13	12	12	14	14

Training: source   target words ⇒ source words

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0	3	5	6	2	1	6	7	11	12	9	16	18	17	15	9

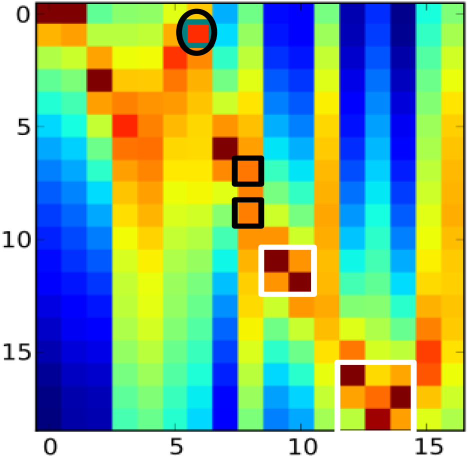
Training: target   target words ⇒ source words

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	3	3	5	2	2	1	6	7	11	12	15	16	18	17	18	14

Alignment

source words	aligned target words(occurrences)
Je	I(4)
vous	you(4)
demands	ask(4), more(1)
donc	therefore(4), would(1)
à	to(1), once(1)
nouveau	once(4)
de	to(4), more(1)
faire	ensure(3), to(1)
le	to(1), ensure(1)
nécessaire	ensure(2), get(1), well(1)
pour	to(1), ensure(1)
que	that(5)
nous	we(4)
puissions	get(1), we(1)
disposer	Durch(1), as(1), well(1)
d'	as(2), get(1), a(1)
une	a(4)
chaîne	channel(4)
néerlandaise	Dutch(3), channel(1), as(1)

Phrase prediction



Je	0	I	0
vous	1	would	1
demands	2	therefore	2
donc	3	once	3
à	4	more	4
nouveau	5	ask	5
de	6	you	6
faire	7	to	7
le	8	ensure	8
nécessaire	9	that	9
pour	10	we	10
que	11	get	11
nous	12	a	12
puissions	13	Dutch	13
disposer	14	channel	14
d’	15	as	15
une	16	well	16
chaîne	17		
néerlandaise	18		

Phrase prediction

- Collect the target words most relating to the words of a given source phrase,
- A target word has edges going into this source phrase and into its complement with respect to the sentence.
- Consider the former as positive edges and the latter ones as negative ones.
- If the sum of scores on the positive edges greater than on the negatives then the word belongs to the translation of the source phrase. Where the score is equal to

$$\mathscr{R}_{\mathbf{W}}(w_s, w_t) \quad = \langle \psi_{\mathcal{P}_{\mathcal{S}_s}}(w_s), \mathbf{W}\phi_{\mathcal{S}_s,\mathcal{S}_t}(w_t) \rangle$$

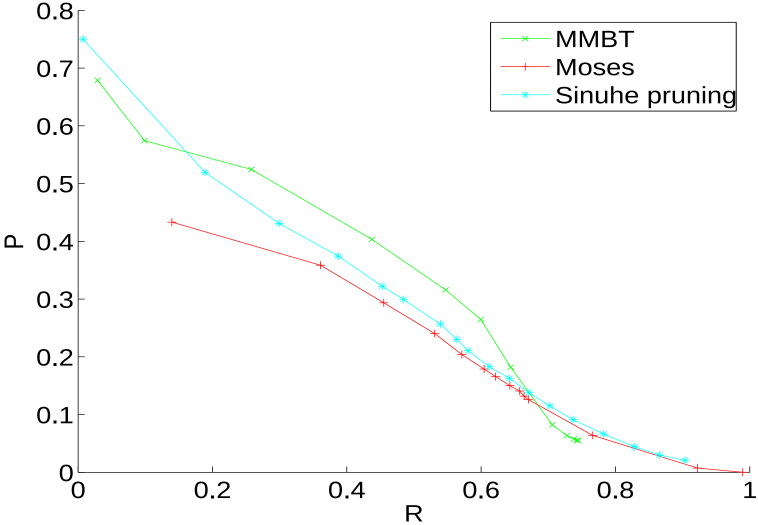
- The gapes can be allowed or prohibited in both side. No gap dependency!
- Phrase score is just the sum of the scores of the words within in the current implementation.

Offline versus online, parallel processing

- The **update of the phrase table** works in **online fashion**, all new sentences are **processed incrementaly.**
- **Computation** of the features, optimization, phrase prediction **can be evaluated parallel** in a multiprocesor system.

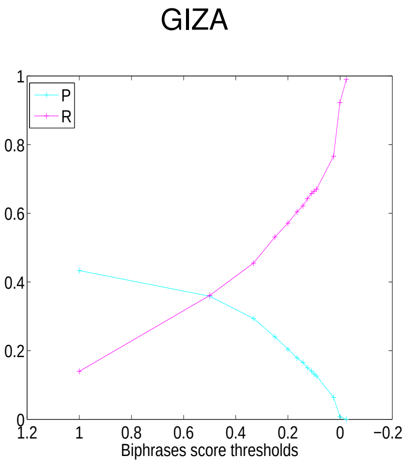
MMBT versus GIZA

Recall versus Precision



MMBT versus GIZA

Tuning Recall and Precision



Current state

- On a desktop machine, CPU: Intel 2.1GHz, $\boxed{\sim\text{5 sentences}}$ per second can be trained assuming the average length of the Europarl sentences.
- The memory requirement is $\boxed{\sim \text{8GB}}$ at a $\boxed{\text{1 million sentence}}$ training corpus, which can be reduced to half on the expense of the speed.
- Accuracies with the decoder to be developed parallel, which currently translates $\boxed{\text{50 sentences/ second}}$ if the phrase dictionary stored in a memory disk:

Languages	Bleu	Nist	Training size/Test size
French-English	**0.2642**	**7.6713**	**1.3million/10000**

- The prototype is written in $\boxed{\text{pure Python code}}$.

This is the End ...

Thanks!