Exploiting Source Similarity for SMT Using Context-Informed Features

Nicolas Stroppa (nstroppa@computing.dcu.ie)

Antal van den Bosch (Antal.vdnBosch@uvt.nl)

Andy Way (away@computing.dcu.ie)

Overview

1. Motivation
2. The Standard Approach
3. Context-Informed Features
4. Memory-Based Disambiguation
5. An Example
6. Evaluation & Results
7. Related Work
8. Conclusions
9. Future Work

Motivation

- SMT is *target-similarity-based*;
- EBMT is *source-similarity-based*.

Can we exploit both benefits in one model?

Motivation: SMT is *target-similarity-based*

The probability of a target sentence w.r.t. an *n*-gram-based LM can be seen as a measure of similarity between this sentence and those sentences found in the training corpus C.

The LM will assign high probabilities to those sentences that share lots of *n*-grams with the sentences in C, while sentences with few *n*-gram matches will receive low probabilities.

$\Rightarrow$ the LM is used to make the resulting translation as similar as possible to previously seen target sentences.

Motivation: EBMT is *source-similarity-based*

There are 3 processing stages in EBMT:

1. retrieving 'similar' fragments of the input string against the reference corpus;
2. identifying the corresponding translation fragments;
3. recombining these translation fragments into the appropriate target text.

Depending on the exact EBMT method used, different notions of 'similarity' are employed.

However, all models of EBMT rely on the retrieval of *source* sentences similar to the new input string in the training material.

Motivation: Benefits of a *Combined* Model

- Source similarity may limit ambiguity problems;
- Target similarity may avoid problems such as *boundary friction*.

By exploiting the two types of similarity, we might benefit from the strengths of both aspects.

Overview

1. Motivation
2. The Standard Approach
3. Context-Informed Features
4. Memory-Based Disambiguation
5. An Example
6. Evaluation & Results
7. Related Work
8. Conclusions
9. Future Work

The Standard Approach: Phrase-Based SMT

In SMT, translation is modeled as a decision process, in which the translation $e_1^I = e_1 \dots e_i \dots e_I$ of a source sentence $f_1^J = f_1 \dots f_j \dots f_J$ is chosen to maximize:

$$\arg\max_{I,e_1^I} P(e_1^I|f_1^J) = \arg\max_{I,e_1^I} P(f_1^J|e_1^I).P(e_1^I) \quad (1)$$

The Standard Approach: Translation Model

$$\arg\max_{I,e_1^I} P(e_1^I|f_1^J) = \arg\max_{I,e_1^I} P(f_1^J|e_1^I).P(e_1^I) \quad (2)$$

The Standard Approach: Language Model

$$\arg\max_{I,e_1^I} P(e_1^I|f_1^J) = \arg\max_{I,e_1^I} P(f_1^J|e_1^I).P(e_1^I) \quad (3)$$

The Standard Approach: Log-linear phrase-based SMT

In log-linear phrase-based SMT, the posterior probability $P(e_1^I|f_1^J)$ is directly modeled as a (log-linear) combination of features [Och & Ney, ACL-02], that usually comprise M translational features (e.g. sentence length, lexical features, grammatical dependencies), and the language model:

$$\log P(e_1^I|f_1^J) = \sum_{m=1}^{m} \lambda_m h_m(f_1^J, e_1^I, s_1^K) + \lambda_{LM} \log P(e_1^I) \qquad (4)$$

where $s_1^K = s_1 \ldots s_k$ denotes a segmentation of the source and target sentences respectively into the sequences of phrases $(\tilde{f}_1, \ldots, \tilde{f}_k)$ and $(\tilde{e_1}, \ldots, \tilde{e_k})$.

The Standard Approach: Log-linear phrase-based SMT

Each feature h_m in log-linear PB-SMT can be rewritten as:

$$h_m(f_1^J, e_1^I, s_1^K) = \sum_{k=1}^{K} \tilde{h}_m(\tilde{f}_k, \tilde{e_k}, s_k), \quad (5)$$

where $\tilde{h}_m$ is a feature that applies to a single phrase-pair.

That is, while the features in log-linear PB-SMT *can* apply to entire sentences in theory, in practice, those features apply to *single phrase pairs* (in existing models).

Remarkably, then, the usual translational features involved in those models only depend on an individual pair of source/target phrases, i.e. they do not take into account the *contexts* of those phrases.

The Standard Approach: Log-linear phrase-based SMT

In this context, the translation process amounts to:

- choosing a segmentation of the source sentence,
- translating each source phrase, and possibly
- re-ordering the target segments obtained.

But translational choices are strongly driven by the target LM.

Instead, we will try to use the source context to resolve ambiguities ...

The Standard Approach: Log-linear phrase-based SMT

Why do we need to try to integrate source language context?

Why can't we just add an LM for the *source* language?

The Standard Approach: Log-linear phrase-based SMT

Why do we need to try to integrate source language context?

Why can't we just add an LM for the *source* language?

$$\arg\max_{I,e_1^I} P(e_1^I|f_1^J) = \arg\max_{I,e_1^I} P(f_1^J|e_1^I).P(e_1^I) \quad (6)$$

The Standard Approach: Log-linear phrase-based SMT

Why do we need to try to integrate source language context?

Why can't we just add an LM for the *source* language?

$$\arg\max_{I,e_1^I} P(e_1^I|f_1^J) = \frac{\arg\max_{I,e_1^I} P(f_1^J|e_1^I).P(e_1^I)}{P(f_1^J)} \quad (7)$$

The Standard Approach: Log-linear phrase-based SMT

Why do we need to try to integrate source language context?

Why can't we just add an LM for the *source* language?

$$\arg\max_{I,e_1^I} P(e_1^I|f_1^J) = \frac{\arg\max_{I,e_1^I} P(f_1^J|e_1^I).P(e_1^I).P(f_1^J)}{P(f_1^J)} \quad (8)$$

The outcome of $\arg\max$ does not change if you add or delete $P(f)$.

Overview

1. Motivation
2. The Standard Approach
3. Context-Informed Features
4. Memory-Based Disambiguation
5. An Example
6. Evaluation & Results
7. Related Work
8. Conclusions
9. Future Work

Context-Informed Features: Disambiguation

C'è una partita di baseball oggi ? $\Leftrightarrow$ *Is there a baseball game today?*

– Possible translations for *partita*:

game	*partita di calcio* $\Leftrightarrow$ *a soccer game*
gone	*è partita* $\Leftrightarrow$ *she has gone*
partita	*una partita di Bach* $\Leftrightarrow$ *a partita of Bach*

– Possible translations for *di*:

of	*una tazza di caffè* $\Leftrightarrow$ *a cup of coffee*
	prima di partire $\Leftrightarrow$ *before coming*

Examples of ambiguity for the (Italian) word *partita*, easily solved when considering its context.

Context-Informed Features: Disambiguation

In standard PB-SMT, disambiguation strongly relies on the *target* LM.

Although the various translation features associated with *partita* and *game*, *partita* and *gone*, etc., depend on the type of training data used, most LMs may still select the correct translation *baseball game* as the most probable among all the possible combinations of target words: *gone of baseball*, *game of baseball*, *baseball partita*, *baseball game*, etc.

If nothing else, this solution is more expensive than simply looking at the *source* context.

In particular, using context can help prune weak candidates early, allowing more time to be spent on more promising candidates.

Context-Informed Features: Discriminative Approaches

Several MT frameworks have been proposed recently to fully exploit the flexibility of discriminative approaches.

Unfortunately, this flexibility usually comes at the price of training complexity.

We pursue an alternative approach: introducing context-informed features *directly* in the original log-linear framework.

In so doing we can take the context of source phrases into account, and still benefit from the existing training and optimization procedures of standard PB-SMT.

Context-Informed Features: Word-Based features

We can use a feature that includes the direct left context and right context words of a given phrase $\tilde{f}_k = f_{b_k} \ldots f_{j_k}$:

$$h_m(f_1^J, e_1^I, s_1^K) = \sum_{k=1}^{K} \tilde{h}_m(\tilde{f}_k, f_{b_k-1}, f_{j_k+1}, \tilde{e_k}, s_k).$$

Context-Informed Features: Word-Based features

We can use a feature that includes the direct left context and right context words of a given phrase $\tilde{f}_k = f_{b_k} \ldots f_{j_k}$:

$$h_m(f_1^J, e_1^I, s_1^K) = \sum_{k=1}^{K} \tilde{h}_m(\tilde{f}_k, f_{b_k-1}, f_{j_k+1}, \tilde{e}_k, s_k).$$

Here, the context is a window of size 3 (focus phrase + left context word + right context word), centred on the source phrase $\tilde{f}_k$.

Larger contexts may also be considered, so more generally, we have:

$$h_m(f_1^J, e_1^I, s_1^K) = \sum_{k=1}^{K} \tilde{h}_m(\tilde{f}_k, CI(\tilde{f}_k), \tilde{e}_k, s_k),$$

where $CI(\tilde{f}_k)$ denotes some contextual information about $\tilde{f}_k$.

Context-Informed Features: Class-Based features

In addition to the context words themselves, it is possible to exploit several knowledge sources characterizing the context.

For example, we can consider the Part-Of-Speech of the focus phrase and of the context words. In our model, the POS of a multi-word focus phrase is the concatenation of the POS tags of the words composing that phrase.

Here, the context for a window of size 3 looks as follows:

$$CI(\tilde{f}_k) = \langle POS(\tilde{f}_k), POS(f_{b_k-1}), POS(f_{j_k+1}) \rangle.$$

We can, of course, combine the class-based and the word-based information together if it leads to further improvements.

Overview

1. Motivation
2. The Standard Approach
3. Context-Informed Features
4. Memory-Based Disambiguation
5. An Example
6. Evaluation & Results
7. Related Work
8. Conclusions
9. Future Work

Memory-Based Disambiguation: Classification

To avoid problems of directly estimating the probabilities required, we use the memory-based classifier IGTREE [Daelemans et al., 97].

More precisely, in order to estimate the probability $P(\tilde{e_k}|\tilde{f}_k, CI(\tilde{f}_k))$, we use IGTREE to classify the input $\langle \tilde{f}_k, CI(\tilde{f}_k) \rangle$.

The result of this classification is a set of weighted class labels, representing the possible target phrases $\tilde{e_k}$.

Once normalized, these weights can be seen as the posterior probabilities of the target phrases $\tilde{e_k}$, which thus gives access to $P(\tilde{e_k}|\tilde{f}_k, CI(\tilde{f}_k))$.

Memory-Based Disambiguation: Classification

To build the set of examples required to train IGTREE, we slightly modify the standard phrase extraction procedure of [Koehn et al., HLT-03] so that we *simultaneously* extract the context information of the source phrases; since these aligned phrases are needed in the standard PB-SMT approach, the context extraction comes at no additional cost.

There are several reasons for using a memory-based classifier such as IGTREE:

- training can be performed efficiently, even with millions of examples,
- it is insensitive to the number of output classes,
- its output can be seen as a posterior distribution.

Overview

1. Motivation
2. The Standard Approach
3. Context-Informed Features
4. Memory-Based Disambiguation
5. An Example
6. Evaluation & Results
7. Related Work
8. Conclusions
9. Future Work

An Example

Given that the features in log-linear PB-SMT apply to single phrase pairs (in existing models), we can build a t-table containing phrase pairs and the values of the features associated with those pairs.

Let's assume the features are $P(\tilde{f}|\tilde{e})$, $P(\tilde{e}|\tilde{f})$, where $\tilde{f}$ and $\tilde{e}$ are source and target phrases respectively.

Let's also assume that the t-table looks like this:

$\tilde{f}$	$\tilde{e}$	$P(\tilde{f}\|\tilde{e})$	$P(\tilde{e}\|\tilde{f})$
the big cat	le grand chat	0.7	0.2
the big cat	le gros chat	0.6	0.8

An Example

Let's now add some (source language) context:

	$\tilde{f}$	$\tilde{e}$	$P(\tilde{f}\|\tilde{e})$	$P(\tilde{e}\|\tilde{f})$
if (context_1)	the big cat	le grand chat	0.7	0.3
if (context_2)	the big cat	le grand chat	0.7	0.1
if (context_1)	the big cat	le gros chat	0.6	0.7
if (context_2)	the big cat	le gros chat	0.6	0.9

That is, the values of $P(\tilde{e}|\tilde{f})$ change depending on the context.

An Example

Question: How can we some up with probabilities that take some context into account?

An Example

Question: How can we some up with probabilities that take some context into account?

Answer: By using our classifiers.

An Example

Question: How can we some up with probabilities that take some context into account?

Answer: By using our classifiers.

Assume the input is the source phrase plus its context (e.g. *the big cat* and its left and right context), and the output classes are the target phrases (*le grand chat, le gros chat*).

Let's ask the classifier: if the possible output classes are *le grand chat* and *le gros chat*, and the input is *the big cat* with the context context_1, which output class (i.e. target phrase) would you pick?

An Example

More precisely, instead of asking the classifier to take a hard decision, we just ask it to assign weights to the possible classes.

To add this new information, we add a feature (i.e. a column in the t-table).

The new t-table becomes:

	$\tilde{f}$	$\tilde{e}$	$P(\tilde{f}\|\tilde{e})$	$P(\tilde{e}\|\tilde{f})$	$P(\tilde{e}\|\tilde{f}+context)$
if ($context_1$)	the big cat	le grand chat	0.7	0.2	0.3
if ($context_2$)	the big cat	le grand chat	0.7	0.2	0.1
if ($context_1$)	the big cat	le gros chat	0.6	0.8	0.7
if ($context_2$)	the big cat	le gros chat	0.6	0.8	0.9

where $P(\tilde{e}|\tilde{f}+context)$ is given by the classifier (a kind of 'pre-decoder').

Overview

1. Motivation
2. The Standard Approach
3. Context-Informed Features
4. Memory-Based Disambiguation
5. An Example
6. Evaluation & Results
7. Related Work
8. Conclusions
9. Future Work

Evaluation & Results: Data

- Chinese–English IWSLT-06;
- Italian–English IWSLT-06.

Data extracted from the *Basic Travel Expressions Corpus* (BTEC) [Takezawa et al., 02].

Multilingual speech corpus containing sentences similar to those usually found in phrase-books for tourists going abroad.

Evaluation & Results: Data Sizes

	Chinese–English	Italian–English
Train.		
Sentences	44,501	21,484
Running words	323,958 351,303	156,237 169,476
Vocabulary size	11,421 10,363	10,418 7,359
Train. examples	434,442	391,626
Dev.		
Sentences	489 (7 refs.)	489 (7 refs.)
Running words	5,214 39,183	4,976 39,368
Vocabulary size	1,137 1,821	1,234 1,776
Test examples	8,004	7,993
Eval.		
Sentences	500 (7 refs.)	500 (7 refs.)
Running words	5,550 44,089	5,787 44,271
Vocabulary size	1,328 2,038	1,467 1,976
Test examples	8,301	9,103

Chinese–English and Italian–English corpus statistics

Evaluation & Results: Training

- Default training sets, plus:
  - devset 1
  - devset 2
  - devset 3
- devset 4 used for tuning, especially for optimising the weights of the log-linear model;
- Evaluation carried out on test sets provided using ‘Correct Recognition Result’ (CRR) input condition;
- for both Italian and Chinese, POS-tagging performed using MXPOST tagger [Ratnaparkhi, EMNLP-96].

Evaluation & Results: Metrics

- BLEU [Papineni et al., ACL-02]
- NIST [Doddington, HLT-02]
- METEOR [Banerjee & Lavie, ACL-05]

For BLEU and NIST, we also computed statistical significance p-values, estimated using approximate randomisation [Noreen, 89].

Evaluation & Results

Used MOSES as Baseline System:

- phrase-based probabilities and lexical weighting in both directions;
- phrase and word penalties;
- reordering

The only additional component is that which avails of our memory-based features.

Evaluation & Results

	BLEU[%] (p-value)	NIST (p-value)	METEOR[%]
Italian–English			
Baseline	37.84	8.33	65.63
POS-only	**38.56** (< 0.1)	8.45 (< 0.02)	66.03
Words-only	37.93 ($\times$)	8.43 (< 0.02)	66.11
Words+POS	38.12 ($\times$)	**8.46** (< 0.01)	**66.14**
Chinese–English			
Baseline	18.81	5.95	47.17
POS-only	19.64 (< 0.005)	6.10 (< 0.005)	47.82
Words-only	**19.86** (< 0.02)	**6.23** (< 0.002)	**48.34**
Words+POS	19.19 ($\times$)	6.09 (< 0.005)	47.97

Italian–English and Chinese–English Translation Results

Evaluation & Results: Remarks

Italian–English:

- Consistent improvement for all metrics, for each type of contextual information: Words-only, POS-only, and Words+POS.
- Compared to baseline, improvements are significant for NIST, and marginally significant (p-value < 0.1) for BLEU only for POS.
- Words + POS leads to slight improvement in METEOR score compared to Words-only and POS-only.
- Best results w.r.t. BLEU score for POS-only. Differences between POS-only, Words-only and Words+POS not statistically significant.

We comment on the differences in significance between BLEU and NIST scores in a few moments.

Evaluation & Results: Remarks

Chinese–English:

- Consistent improvement for all metrics, for each type of contextual information.
- Compared to baseline, improvements are significant for NIST for Words-only, POS-only and Words+POS.
- W.r.t. BLEU score, adding Words+POS not useful: Words-only and POS-only scores are much higher than Words+POS. This is due to poor quality tagging – tagging accuracy for Italian is qualitatively higher.

Evaluation & Results: Feature Information Gain

	Italian–English		Chinese–English	
Rank	Feature	IG	Feature	IG
1	W(0)	7.82	W(0)	6.74
2	P(0)	4.59	W(+1)	3.73
3	W(+1)	4.24	P(0)	3.23
4	W(-1)	4.09	W(-1)	3.21
5	W(+2)	3.19	W(+2)	2.90
6	W(-2)	2.84	W(-2)	2.25
7	P(+1)	1.75	P(-1)	1.18
8	P(-1)	1.61	P(+1)	1.03
9	P(-2)	0.94	P(-2)	0.77
10	P(+2)	0.90	P(+2)	0.75

- Word information $>$ POS information
- Focus $>$ Right context $>$ Left context
- +/- 1 $>$ =/-2

Evaluation & Results: Statistical Significance

Since BLEU and NIST are both *n*-gram-based metrics, it might be seen as strange that improvements may be statistically significant for NIST, but insignificant for BLEU.

The differences between the two metrics are:

- max. length of *n*-gram considered (4 for BLEU, 5 for NIST);
- weighting of the matched *n*-grams (none for BLEU, information-based weighting for NIST);
- type of mean used to aggregate the number of matched *n*-grams for different *n* (geometric for BLEU, arithmetic for NIST);
- length penalty.

Evaluation & Results: Statistical Significance

For the 16 (2^4) combinations of these differences, for the three cases where there was a disagreement w.r.t. statistical significance between BLEU and NIST, the most important factors were:

- information-based weighting;
- type of mean used.

BLEU's geometric mean tends to ignore good lexical changes, whereas the information-based weighting favours the most difficult lexical choices.

These findings are consistent with those of [Riezler & Maxwell, ACL-05].

Overview

1. Motivation
2. The Standard Approach
3. Context-Informed Features
4. Memory-Based Disambiguation
5. An Example
6. Evaluation & Results
7. Related Work
8. Conclusions
9. Future Work

Related Work

Discriminative Learning:

- Cowan et al., EMNLP-06;
- Liang et al., COLING-ACL-06;
- Tillmann & Zhang, COLING-ACL-06;
- Wellington et al., AMTA-06.

In general, these papers require one's training procedures to be redefined.

Our approach introduces new features, yet maintains the strengths of existing state-of-the-art systems.

Related Work

Combining EBMT & SMT:

- Groves & Way, ACL-05, 2006.

Combining both ‘SMT-style’ and ‘EBMT-style’ chunks in a hybrid system.

Word-Sense Disambiguation

- Carpuat & Wu, EMNLP-07, TMI-07!!

WSD techniques enhance lexical selection.

We’re doing something similar, yet totally *implicitly*.

Overview

1. Motivation
2. The Standard Approach
3. Context-Informed Features
4. Memory-Based Disambiguation
5. An Example
6. Evaluation & Results
7. Related Work
8. Conclusions
9. Future Work

Conclusions

- introduced new features for log-linear phrase-based SMT, that take into account contextual information from the *source* language;
- presented a memory-based classification framework that enables the estimation of these features while avoiding sparseness problems;
- reported significant improvements for both BLEU and NIST scores when adding these context-informed features on Italian-to-English and Chinese-to-English translation tasks.

Overview

1. Motivation
2. The Standard Approach
3. Context-Informed Features
4. Memory-Based Disambiguation
5. An Example
6. Evaluation & Results
7. Related Work
8. Conclusions
9. Future Work

Future Work

1. investigate the addition of features including syntactic information;
2. try different taggers;
3. introduce context-informed lexical smoothing features, similarly to the standard phrase-based approach;
4. modify the decoder to directly integrate context-informed features;
5. directly compare the hybrid system of [Groves & Way, 05, 06] to this work.

Questions

Thanks for listening!