Exploiting Source Similarity for SMT using Context-Informed Features

Nicolas Stroppa
Dublin City University,
Dublin,
Ireland
nstroppa@computing.dcu.ie

Antal van den Bosch
Tilburg University,
Tilburg,
The Netherlands
Antal.vdnBosch@uvt.nl

Andy Way
Dublin City University,
Dublin,
Ireland
away@computing.dcu.ie

Abstract

In this paper, we introduce context-informed features in a log-linear phrase-based SMT framework; these features enable us to exploit source similarity in addition to target similarity modeled by the language model. We present a memory-based classification framework that enables the estimation of these features while avoiding sparseness problems. We evaluate the performance of our approach on Italian-to-English and Chinese-to-English translation tasks using a state-of-the-art phrase-based SMT system, and report significant improvements for both BLEU and NIST scores when adding the context-informed features.

1 Introduction

In log-linear phrase-based SMT, the probability $\mathbb{P}(e_1^I|f_1^J)$ of target phrase e_1^I given a source phrase f_1^J is modeled as a (log-linear) combination of features that usually comprise some translational features, and a language model (Och and Ney, 2002). The usual translational features involved in those models express dependencies between source and target phrases, but not dependencies between source phrases themselves. In particular, the context in which those phrases occur is never taken into account during translation. While the language model can be seen as a way to exploit *target similarity* (between the translation and other target sentences), one could ask whether it is also possible to exploit *source similarity*, i.e. to take into account the context in which the source phrases to be translated actually occur.

In this paper, we introduce context-informed features in the original log-linear model, enabling us to take the context of source phrases into account during translation. In order to tackle the problems related to the estimation of these features, we propose a framework based on a memory-based classifier, which performs implicit smoothing. We also show that the addition of context-informed features, i.e. the source-similarity exploitation, results in an improvement in translation quality, for Italian-to-English and Chinese-to-English translations tasks.

2 Log-Linear Phrase-Based SMT

In statistical machine translation (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 $f_1^J = f_1^J = f_1 \dots f_j \dots f_J$ is chosen to maximize:

$$\underset{I,e_1^I}{\operatorname{argmax}}\, \mathbb{P}(e_1^I|f_1^J) = \underset{I,e_1^I}{\operatorname{argmax}}\, \mathbb{P}(f_1^J|e_1^I).\mathbb{P}(e_1^I), \quad (1)$$

where $\mathbb{P}(f_1^J|e_1^I)$ and $\mathbb{P}(e_1^I)$ denote respectively the translation model and the target language model (Brown et al., 1993). In log-linear phrase-based SMT, the posterior probability $\mathbb{P}(e_1^I|f_1^J)$ is directly modeled as a (log-linear) combination of features (Och and Ney, 2002), that usually comprise M translational features, and the language

model:

$$\log \mathbb{P}(e_1^I|f_1^J) = \sum_{m=1}^{m} \lambda_m h_m(f_1^J, e_1^I, s_1^K) + \lambda_{LM} \log \mathbb{P}(e_1^I), \quad (2)$$

where $s_1^K = s_1 \dots s_k$ denotes a segmentation of the source and target sentences respectively into the sequences of phrases $(\tilde{e_1}, \dots, \tilde{e_k})$ and $(\tilde{f_1}, \dots, \tilde{f_k})$ such that (we set $i_0 := 0$):

$$\forall 1 \leq k \leq K, \quad s_k := (i_k; b_k, j_k),$$
$$\tilde{e_k} := e_{i_{k-1}+1} \dots e_{i_k},$$
$$\tilde{f_k} := f_{b_k} \dots f_{j_k}.$$

A remarkable property of this approach is that the usual translational features involved in those models only depend on a pair of source/target phrases, i.e. they do not take into account the contexts of those phrases. This means that each feature h_m in equation (2) can be rewritten as:

$$h_m(f_1^J, e_1^I, s_1^K) = \sum_{k=1}^{K} \tilde{h}_m(\tilde{f_k}, \tilde{e_k}, s_k), \quad (3)$$

where $\tilde{h}_m$ is a feature that applies to a single phrase-pair.[1] It thus follows:

$$\sum_{m=1}^{m} \lambda_m \sum_{k=1}^{K} \tilde{h}_m(\tilde{f_k}, \tilde{e_k}, s_k) = \sum_{k=1}^{K} \tilde{h}(\tilde{f_k}, \tilde{e_k}, s_k),$$
$$\text{with } \tilde{h} = \sum_{m=1}^{m} \lambda_m \tilde{h}_m. \quad (4)$$

In this context, the translation process amounts to: (i) choosing a segmentation of the source sentence, (ii) translating each source phrase, and possibly (iii) re-ordering the target segments obtained. The target language model is used to guide the decision process; in case no particular constraints are assumed, it is common to employ beam search techniques to reduce the number of hypotheses to be considered (Koehn, 2004). Equations (2) and (4) characterize what is referred to as the *standard phrase-based approach* in the following.

[1] Here, for notational purposes, we exclude re-ordering features that might not be expressed using equation (3). This does not affect our general line of reasoning.

C'è una partita di baseball oggi ?
(⇔ *Is there a baseball game today?*)

– Possible translations for *partita*:

game	*partita di calcio* ⇔ *a soccer game*
gone	*è partita* ⇔ *she has gone*
partita	*una partita di Bach* ⇔ *a partita of Bach*

– Possible translations for *di*:

of	*una tazza di caffè* ⇔ *a cup of coffee*
	prima di partire ⇔ *before coming*

Figure 1: Examples of ambiguity for the (Italian) word *partita*, easily solved when considering its context

3 Context-Informed Features

3.1 Context-Based Disambiguation

The optimization of the feature weights λ_m can be performed in a *discriminative* learning setting (Och and Ney, 2002). However, it is important to note that these weights are *meta-parameters*. Indeed, the dependencies between the parameters of the standard phrase-based approach consist of: (i) relationships between single phrases (modeled by $\tilde{h}$), (ii) relationships between consecutive target words (modeled by the language model), which is generally characteristic of *generative* models (Collins, 2002; Dietterich, 2002). Notably, dependencies between consecutive *source* phrases are not directly expressed.

Discriminative frameworks usually allow for the introduction of (relatively) unrestricted dependencies that are relevant to the decision process. In particular, disambiguation problems can be solved by taking the direct context of the entity to disambiguate into account (e.g. Dietterich (2002)). In the translation example displayed in Figure 1, the source right context is sufficient to solve the ambiguity: when followed by *di baseball*, the (Italian) word *partita* is very likely to correspond to the (English) word *game*.

However, in the standard phrase-based approach, the disambiguation strongly relies on the *target* language model. Indeed, even though the various translation features associated with *partita* and *game*, *partita* and *gone*, etc., may depend on the type of data on which the model is trained, it is likely that most language models will 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., but this solution appears to be more expensive than simply looking at the context. In particular, the context can be used to early prune weak candidates, which allows spending more time on promising candidates.

Several discriminative frameworks have been proposed recently in the context of MT to fully exploit the flexibility of discriminative approaches (Cowan et al., 2006; Liang et al., 2006; Tillmann and Zhang, 2006; Wellington et al., 2006). Unfortunately, this flexibility usually comes at the price of training complexity. An alternative in-between approach, pursued in this paper, consists of introducing context-informed features in the original log-linear framework. This enables us to take the context of source phrases into accounts, while benefiting from the existing training and optimization procedures of the standard phrase-based approach.

3.2 Context-Informed Features

In this Section, we introduce several features that take the context of source phrases into account.

Word-based features A feature that includes the direct left and right context words (resp. f_{b_k-1} and f_{j_k+1}) of a given phrase $\tilde{f}_k = f_{b_k} \dots f_{j_k}$ takes the following form:

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

In this case, the contextual information can be seen as a window of size 3 (focus phrase + left context word + right context word), centered on the source phrase $\tilde{f}_k$. Larger contexts may also be considered. 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 $\tilde{f}_k$.[2]

[2]The definition of the context may be language dependent. For example, one could consider only the right context if it makes sense to do so for a particular language; the same remark holds for the size of the context.

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.[3] In this case, the contextual information takes the following form for a window of size 3:

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

We can also combine the class-based and the word-based information.

Feature definition One natural definition to express a context-informed feature consists of viewing it as the conditional probability of the target phrase given the source phrase and its context information:

$$\tilde{h}_m(\tilde{f}_k, CI(\tilde{f}_k), \tilde{e_k}, s_k) = \log \mathbb{P}(\tilde{e_k} | \tilde{f}_k, CI(\tilde{f}_k)).$$

The problems related to the estimation of these probabilities are addressed in the next section.

4 Memory-Based Disambiguation

4.1 A Classification Approach

The direct estimation of $\mathbb{P}(\tilde{e_k} | \tilde{f}_k, CI(\tilde{f}_k))$, for example using relative frequencies, is problematic. Indeed, it is well known that the estimation of $\mathbb{P}(\tilde{e_k} | \tilde{f}_k)$ using relative frequencies results in the overestimation of the probabilities of long phrases (Zens and Ney, 2004; Foster et al., 2006); a frequent remedy consists of introducing a smoothing factor, which takes the form of lexical-based features (Zens and Ney, 2004). Similar issues and a variety of smoothing techniques are discussed in (Foster et al., 2006). In the case of context-informed features, since the context is also taken into account, this estimation problem can only worsen, which forbids us to use relative frequencies.

To avoid these issues, we use a memory-based classifier, which enables *implicit smoothing*. More precisely, in order to estimate the probability $\mathbb{P}(\tilde{e_k} | \tilde{f}_k, CI(\tilde{f}_k))$, we ask a memory-based classifier to classify the input $\langle \tilde{f}_k, CI(\tilde{f}_k) \rangle$ (seen

[3]The POS of a multi-word focus phrase is the concatenation the POS of the words composing the phrase.

as a fixed-length vector). 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 $\mathbb{P}(\tilde{e_k}|\tilde{f_k}, CI(\tilde{f_k}))$.

In order to build the set of examples required to train the classifier, we slightly modify the standard phrase extraction procedure described in (Koehn et al., 2003) so that it also extracts the context information of the source phrases; since these aligned phrases are needed in the standard phrase-based approach, the context extraction comes at no additional cost.

Note that there are several reasons for using a memory-based classifier: (i) training can be performed efficiently, even with millions of examples, (ii) it is insensitive to the number of output classes, (iii) its output can be seen as a posterior distribution.

4.2 IGTree Classification

In the following, we describe IGTREE,[4] an algorithm for the top-down induction of decision trees that can be seen as an approximation of 1-nearest neighbor that stores and classifies examples efficiently (Daelemans et al., 1997). IGTREE compresses a database of labeled examples into a lossless-compression decision-tree structure that preserves the labeling information of all examples (and technically should be named a *trie* according to Knuth (1973)). In our case, a labeled example is a fixed-length feature-value vector representing the source phrase and its contextual information, associated with a symbolic class label representing the associated target phrase. The trie that is constructed can then be used to predict a target phrase given a source phrase and its context. A typical trie is composed of nodes that each represent a partition of the original example database, together with the most frequent class of that partition. The root node of the trie thus represents the entire example database and carries the most frequent value as class label, while end nodes (leaves) represent a homogeneous partition of the database in which all examples have the same class label. A node is either a leaf or a non-ending node that branches out to nodes at a deeper level of the trie. Each branch represents a test on a feature value; branches fanning out of one node test on values of the same feature.

Prediction in IGTREE is a straightforward traversal of the trie from the root node down, where a step is triggered by an exact match between a feature of the new example and an arc fanning out of the current node. When the next step ends in a leaf node, the homogeneous class at that node is returned; when no match is found with an arc fanning out of the current node, the most likely class stored at that node is returned.

To attain high compression levels, IGTREE adopts the same heuristic that most other decision-tree induction algorithms adopt, such as C4.5 (Quinlan, 1983), which is to create trees from a starting root node and branch out to test on the most informative, or most class-discriminative features first. Like C4.5, IGTREE uses information gain (IG) to estimate the discriminative power of features. The key difference between IGTREE and C4.5 is that IGTREE computes the IG of all features once on the full database of training examples, makes a feature ordering once on these computed IG values, and uses this ordering throughout the whole trie. Moreover, IGTREE does not prune its produced trie, so that it performs a lossless compression of the labeling information of the original example database. In case of exact matches, the exact same output will be retrieved.

IGTREE bases its classification on the example that matches on most features, ordered by their IG, and guesses a majority class of the set of examples represented at the level of mismatching. In our case, we do not keep just the majority class since we want to be able to estimate $\mathbb{P}(\tilde{e_k}|\tilde{f_k}, CI(\tilde{f_k}))$ for all possible $\tilde{e_k}$; we are thus interested in the entire set of labels represented at the level of mismatching. Each possible target phrase can be supported by multiple votes, which leads to a weighted set of target phrases. By normalizing these weights, we obtain the posterior probability distributions we are interested in.[5]

[4] An implementation of IGTree is freely available as part of the TiMBL software package, which can be downloaded from `http://ilk.uvt.nl/timbl`.

[5] It is also interesting to note that if we do not include any context information, the (normalized) output provided by IGTREE exactly corresponds to the conditional probabil-

4.3 Memory-Based Features

The weighted set of possible target phrases given a source phrase and its context is an intermediary result of the estimation of $\mathbb{P}(\tilde{e_k}|\tilde{f_k}, CI(\tilde{f_k}))$. In addition to the feature $\tilde{h}_m(\tilde{f_k}, CI(\tilde{f_k}), \tilde{e_k}, s_k) = \log \mathbb{P}(\tilde{e_k}|\tilde{f_k}, CI(\tilde{f_k}))$, we consider a simple binary feature based on this intermediary result:

$$\tilde{h}_{best} = \begin{cases} 1 & \text{if } \tilde{e_k} \text{ is (one of) the target phrases} \\ & \text{with the most support,} \\ 0 & \text{otherwise,} \end{cases}$$

where "most support" means the highest probability according to $\mathbb{P}(\tilde{e_k}|\tilde{f_k}, CI(\tilde{f_k}))$. The two features $\tilde{h}_m$ and $\tilde{h}_{best}$ are integrated in the log-linear model. As for the standard phrase-based approach, their weights are optimized using minimum-error-rate training (Och, 2003).

4.4 Implementation Issues

When predicting a target phrase given a source phrase and its context, the source phrase is intuitively the feature with the highest prediction power; in all our experiments, it is the feature with the highest IG. In the trie constructed by IGTREE, this is thus the feature on which the first branching decision is taken. Consequently, when classifying a source phrase $\tilde{f_k}$ with its context, there are two possible situations, depending on $\tilde{f_k}$ being in the training material or not. In the first case, $\tilde{f_k}$ is matched, and we proceed further down the trie. At this stage, it follows that the target phrases that can be retrieved are only those that have been aligned to $\tilde{f_k}$. In the second case, $\tilde{f_k}$ cannot be matched, so the full set of labeled leaves of the entire trie is retrieved. Since the second case does not present any interest, we limit the classification to the source phrases contained in the training material. By limiting ourselves to the first situation, we ensure that only target phrases $\tilde{e_k}$ that have been aligned with $\tilde{f_k}$ will be retrieved. This is a desirable property that may be not be necessarily verified if we were using a different type of classifier, more prone to over-generalisation issues.[6]

Phrase-based SMT decoders such as (Koehn, 2004) rely on a phrase-table represented as a list of aligned phrases accompanied with several features. Since these features do not express the context in which those phrases occur, no context information is kept in the phrase-table, and there is no way to recover this information from the phrase-table. In order to take into account the context-informed features with this kind of decoders, we use the workaround described in what follows. Each word to be translated (i.e. appearing in the test set) is assigned a unique id, and each phrase to be translated which is also present in the phrase-table is given to IGTREE for classification. We merge the initial information of the phrase-table concerning this source phrase with the output for IGTREE, to obtain a new phrase-table containing the standard and the context-informed features. In this new phrase-table, each source phrase is represented as a sequence of ids (of the words composing the phrase). By replacing all the words by their ids in the test set, we can translate it using this new phrase-table.

4.5 Source vs. Target Similarity

SMT and target-based similarity The probability of a (target) sentence with respect to a n-gram-based language model can be seen as a measure of similarity between this sentence and the sentences found in the corpus C on which the language model is trained. Indeed, the language model will assign high probabilities to those sentences which share lots of n-grams with the sentences of C, while sentences with few n-grams matches will be assigned low probabilities. In other words, the language model is used to make the resulting translation similar to previously seen (target) sentences: SMT is *target-similarity* based.

EBMT and source-based similarity In order to perform the translation of a given sentence f, Example-Based Machine Translation (EBMT) systems (i) look for source sentences similar to f in the bilingual corpus (retrieval), (ii) find use-

ities $\mathbb{P}(\tilde{e_k}|\tilde{f_k})$ estimated with relative frequencies on the set of aligned phrases.

[6] From the point of view of the classification task, the set of class labels is the set of *all* the target phrases encountered in the training data. Consequently, given a source phrase $\tilde{f_k}$ there is in the general case nothing preventing a classifier to output a target phrase $\tilde{e_k}$ that was never aligned to $\tilde{f_k}$. If we use IGTree and if the source phrase is the feature with the highest information gain, then we have the mentioned desirable property.

ful fragments in these sentences (matching), (iii) adapts and recombine the translation of these fragments (transfer) (Nagao, 1984; Somers, 1999; Carl and Way, 2003). A number of matching techniques and notions of similarity have been proposed. Consequently, EBMT crucially relies on the retrieval of *source* sentences *similar* to f in the bilingual training corpus; in other words, EBMT is *source-similarity* based. Let us also mention (Somers et al., 1994), which marks the fragments to translate with their (left and right) contexts.

Source and Target Similarity While the use of target-similarity may avoid problems such as boundary-friction usually encountered in EBMT (Brown et al., 2003), the use of source-similarity may limit ambiguity problems (cf. Section 3). By exploiting the two types of similarity, we hope to benefit from the strength of both aspects.

5 Experimental Results

5.1 Data, Tasks, and Baseline

The experiments were carried out using the Chinese–English and Italian–English datasets provided within the IWSLT 2006 evaluation campaign (Paul, 2006), extracted from the Basic Travel Expression Corpus (BTEC) (Takezawa et al., 2002). This multilingual speech corpus contains sentences similar to those that are usually found in phrase-books for tourists going abroad. Training was performed using the default training set, to which we added the sets devset1, devset2, and devset3. The development set (devset 4) was used for tuning purposes (in particular for the optimisation of the weights of the log-linear model), and the final evaluation is conducted using the test set (using the CRR=Correct Recognition Result input condition). For both Chinese and Italian, POS-tagging is performed using the MXPOST tagger (Ratnaparkhi, 1996). Table 1 summarizes the various corpus statistics. The number of training/test examples refers to the examples involved in the classification task.

For all experiments, the quality of the translation output is evaluated using the accuracy measures BLEU (Papineni et al., 2002), NIST (Doddington, 2002), and METEOR (Banerjee and Lavie, 2005), using 7 references and ignoring case information. For BLEU and NIST, we also report statistical significance p-values, estimated using approximate randomization (Noreen, 1989; Riezler and Maxwell, 2005).[7]

	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	

Table 1: Chinese–English and Italian–English corpus statistics

To assess the validity of our approach, we use the state-of-the-art phrase-based SMT system MOSES (Koehn et al., 2007).[8] The baseline system is composed of the usual features: phrase-based probabilities and lexical weighting in both directions, phrase and word penalties, and re-ordering. Our system additionally includes the memory-based features described in Sections 3 and 4.

5.2 Translation Results

The results obtained for the Italian–English and Chinese–English translation tasks using the IWSLT data are summarized in Table 2. The contextual information may include the (context) words, their Part-Of-Speech, or both, respectively denoted by Words-only, POS-only, and Words+POS in the following. In all cases, the size of the left context is 2 and so is the size of the right context.[9]

In the case of Italian–English, a consistent improvement is observed for all metrics, for the three types of contextual information (Words-only, POS-only, Words+POS). Relatively to the baseline results, this improvement is significant

[7] The code for statistical significance testing can be freely downloaded from `http://www.computing.dcu.ie/~nstroppa/softs/fast_mt_eval.tgz`.

[8] `http://www.statmt.org/moses/`

[9] These are the values which led to the best results on the development set during the exploratory phase.

	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

Table 2: Italian–English and Chinese–English Translation Results

for NIST, and marginally significant for BLEU (p-value < 0.1) for POS-only. The combination of the words and POS information leads to a slight improvement for NIST and METEOR relatively to Words-only and POS-only. As for the BLEU score, the best results are obtained with POS-only. The difference between POS-only, Word-only, and Words+POS is never statistically significant. The difference of significance between the BLEU and NIST scores is investigated in more depth in Section 5.3.

In the case of Chinese–English, the improvement is also consistent for all metrics, and significant for both BLEU and NIST for Words-only, POS-only, and Words+POS. Interestingly, the addition of Part-of-Speech information does not seem to be benefitial in the case of Chinese. Indeed, the results of Words-only are higher than those obtained with both POS-only and Words+POS. In order to understand better why this is the case, we manually inspected the tagger's output for the Chinese data. The most obvious explanation is simply the (poor) quality of tagging. Indeed, we found lots of tagging mistakes, which contributes to the introduction of noise in the data. We also manually checked that in the case of Italian, the tagging accuracy is qualitatively higher. Consequently, even if there is something to be gained from the addition of POS information, it seems important to ensure that the accuracy of tagging is high enough. Also, with larger training data, it may be sufficient to rely on the words only, since the need for generalization is less important in this case.

In order to know the contribution of the various contextual elements, we rank the contextual features of the Words+POS model based on their Information Gain (cf. Table 3). W(0) and P(0) denotes the focus phrase and its POS, while W(i) and P(i) denotes the word and the POS of the words at position i relative to the focus phrase. The rankings for Italian and Chinese are globally

	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

Table 3: Feature Information Gain

similar, and we can observe the following tendencies:

Word information $>$ POS information,
Focus $>$ Right context $>$ Left context.

5.3 Statistical Significance for n-gram Based Metrics

Since the BLEU and NIST metrics are both precision- and n-gram-based (Doddington, 2002), it is somehow strange that an improvement may be statistically significant for NIST and insignificant for BLEU (as it is the case 3 times in Table 2). The differences between the two metrics are: (i) the maximum length of the n-gram considered (4 for BLEU, 5 for NIST), (ii) the weighting of the matched n-grams

(no weighting for BLEU, information-based weighting for NIST), (iii) the type of mean used to aggregate the number of matched n-grams for different n (geometric for BLEU, arithmetic for NIST), (iv) the length penalty.

To test which of these options were responsible for the difference in significance, we created the 2^4 metrics corresponding to all the possible combinations of options, and we ran the significance tests for the three cases for which there was a disagreement between BLEU and NIST with respect to significance. We found out that the most important factors are the information-based weighting, and the type of mean used. This is actually consistent with our expectation for our system regarding lexical selection. Indeed, BLEU's geometric mean tends to ignore good lexical changes, which may be shadowed by low n-grams results for high values of n; similarly, the information-based weighting favors the most difficult lexical choices. Note that these remarks are also consistent with the findings of (Riezler and Maxwell, 2005).

6 Related Work

Several proposals have been recently made to fully exploit the accuracy and the flexibility of discriminative learning (Cowan et al., 2006; Liang et al., 2006; Tillmann and Zhang, 2006; Wellington et al., 2006). These papers generally require one to redefine one's training procedures; on the contrary our approach introduces new features while keeping the strength of existing state-of-the-art systems. The exploitation of source-similarity is one of the key components of EBMT (Nagao, 1984; Somers, 1999; Carl and Way, 2003); one could say that our approach is a combination of EBMT and SMT since we exploit both source similarity and target similarity. (Carpuat and Wu, 2005) present an attempt to use word-sense disambiguation techniques to MT in order to enhance lexical selection; in a sense, we are also performing some sort of word-sense disambiguation, even if the handling of lexical selection is performed totally implicitly in our case.

7 Conclusion

In this paper, we have introduced new features for log-linear phrase-based SMT, that take into account contextual information about the source phrases to translate. This contextual information can take the form of left and right context words, as well as other source of knowledge such as Part-Of-Speech information. We presented a memory-based classification framework that enables the estimation of these features while avoiding sparseness problems.

We have evaluated the performance of our approach by measuring the influence of the addition of these context-informed features on Italian-to-English and Chinese-to-English translation tasks, using a state-of-the-art phrase-based SMT system. We report significant improvements for both BLEU and NIST scores.

As for future work, we plan to investigate the addition of features including syntactic information. For example, one could consider dependency relationships between the words within the focus (source) phrase or with its close context. We could also introduce context-informed lexical smoothing features, similarly to the standard phrase-based approach. Finally, we plan to modify the decoder to directly integrate context-informed features.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the the ACL Workshop on Intrisic and Extrinsic Evaluation Measures for MT and/or Summarization*, pages 65–72, Ann Arbor, MI.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Ralf D. Brown, Rebecca Hutchinson, Paul N. Bennett, Jaime G. Carbonell, and Peter Jansen. 2003. Reducing boundary friction using translation-fragment overlap. In *Proceedings of the 9th Machine Translation Summit*, pages 24–31, New Orleans, LA.

Michael Carl and Andy Way, editors. 2003. *Recent Advances in Example-Based Machine Translation*, volume 21 of *Text, Speech and Language Technology*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Marine Carpuat and Dekai Wu. 2005. Word sense disambiguation vs. statistical machine translation. In *Proceedings of ACL 2005*, pages 387–394, Ann Arbor, MI.

Michael Collins. 2002. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *Proceedings of EMNLP 2002*, pages 1–8, Philadelphia, PA.

Brooke Cowan, Ivona Kucerovà, and Michael Collins. 2006. A discriminative model for tree-to-tree translation. In *Proceedings of EMNLP 2006*, pages 232–241, Sydney, Australia.

Walter Daelemans, Antal Van den Bosch, and A. Weijters. 1997. IGTree: using trees for compression and classification in lazy learning algorithms. *Artificial Intelligence Review*, 11:407–423.

Thomas G. Dietterich. 2002. Machine learning for sequential data: A review. In Terry Caelli, Adnan Amin, Robert P. W. Duin, Mohamed S. Kamel, and Dick de Ridder, editors, *Structural, Syntactic, and Statistical Pattern Recognition*, volume 2396 of *Lecture Notes in Computer Science*, pages 15–30. Springer-Verlag.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram cooccurrence statistics. In *Proceedings of HLT 2002*, pages 128–132, San Diego, CA.

George Foster, Roland Kuhn, and Howard Johnson. 2006. Phrasetable smoothing for statistical machine translation. In *Proceedings of EMNLP 2006*, pages 53–61, Sydney, Australia.

Donald E. Knuth. 1973. *The art of computer programming*, volume 3: Sorting and searching. Addison-Wesley, Reading, MA.

Philip Koehn, Franz Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT-NAACL 2003*, pages 48–54, Edmonton, Canada.

P. Koehn, M. Federico, W. Shen, N. Bartoldi, O. Bojar, C. Callison-Burch, B. Cowan, C. Dyer, H. Hoang, R. Zens, A. Constantin, C. Moran, and E. Herbst. 2007. Open source toolkit for statistical machine translation: Factored translation models and confusion network decoding. Technical report, Final Report of the Johns Hopkins 2006 Summer Workshop.

Philip Koehn. 2004. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *Proceedings of AMTA 2004*, pages 115–124, Washington, DC.

Percy Liang, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar. 2006. An end-to-end discriminative approach to machine translation. In *Proceedings of COLING-ACL 2006*, pages 761–768, Sydney, Australia.

Makoto Nagao. 1984. A framework of a mechanical translation between Japanese and English by analogy principle. In Alick Elithorn and Ranan Banerji, editors, *Artificial and Human Intelligence*, pages 173–180, Amsterdam, The Netherlands. North-Holland.

Eric W. Noreen. 1989. *Computer-Intensive Methods for Testing Hypotheses: An Introduction*. Wiley-Interscience, New York, NY.

Franz Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of ACL 2002*, pages 295–302, Philadelphia, PA.

Franz Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of ACL 2003*, pages 160–167, Sapporo, Japan.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL 2002*, pages 311–318, Philadelphia, PA.

Michael Paul. 2006. Overview of the IWSLT 2006 Evaluation Campaign. In *Proceedings of IWSLT 2006*, pages 1–15, Kyoto, Japan.

Ross Quinlan. 1983. Learning efficient classification procedures and their application to chess endgames. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*, pages 463–482. Morgan Kaufmann Publishers, Los Altos, CA.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of EMNLP 1996*, pages 133–142, Philadelphia, PA.

Stefan Riezler and John Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing for mt. In *Proceedings of the the ACL Workshop on Intrisic and Extrinsic Evaluation Measures for MT and/or Summarization*, pages 57–64, Ann Arbor, MI.

Harold Somers, Ian McLean, and Danny Jones. 1994. Experiments in multilingual example-based generation. In *CSNLP 1994: 3rd Conference on the Cognitive Science of Natural Language Processing*, Dublin, Ireland.

Harold Somers. 1999. Review article: Example-based machine translation. *Machine Translation*, 14(2):113–157.

T. Takezawa, E. Sumita, F. Sugaya, H. Yamamoto, and S. Yamamoto. 2002. Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world. In *Proceedings of LREC 2002*, pages 147–152, Las Palmas, Spain.

Christoph Tillmann and Tong Zhang. 2006. A discriminative global training algorithm for statistical mt. In *Proceedings of COLING-ACL 2006*, pages 721–728, Sydney, Australia.

Benjamin Wellington, Joseph Turian, Chris Pike, and I. Dan Melamed. 2006. Scalable purely-discriminative training for word and tree transducers. In *Proceedings of AMTA 2006*, pages 251–260, Cambridge, MA.

Richard Zens and Hermann Ney. 2004. Improvements in phrase-based statistical machine translation. In *Proceedings of HLT-NAACL 2004*, pages 257–264, Boston, MA.