Learning Labelled Dependencies in Machine Translation Evaluation

Yifan He and **Andy Way**
Centre for Next Generation Localisation
School of Computing
Dublin City University
{yhe,away}@computing.dcu.ie

Abstract

Recently novel MT evaluation metrics have been presented which go beyond pure string matching, and which correlate better than other existing metrics with human judgements. Other research in this area has presented machine learning methods which learn directly from human judgements. In this paper, we present a novel combination of dependency- and machine learning-based approaches to automatic MT evaluation, and demonstrate greater correlations with human judgement than the existing state-of-the-art methods. In addition, we examine the extent to which our novel method can be generalised across different tasks and domains.

1 Introduction

There is no doubt that the onset of automatic evaluation metrics such as BLEU (Papineni et al., 2002) has led directly to improvements in quality in machine translation (MT). Prior to their introduction, most results were anecdotal, or researchers had to conduct expensive human evaluations in order to validate their work.

However, seven years after their introduction, there is widespread recognition in MT that these string-based metrics are not discriminative enough to reflect the translation quality of today's systems, many of which have gone beyond *n*-grams (cf. (Callison-Burch et al., 2006)).

With that in mind, a number of researchers have come up with metrics which are not wholly string-based. Perhaps the best-known alternative metric is METEOR (Banerjee and Lavie, 2005), which while still being string-based, tries to improve on the matching schemes of BLEU by incorporating synonym matching via WordNet.

Given that many of today's MT systems incorporate some kind of syntactic information (e.g. (Chiang, 2005)), it was perhaps natural that other researchers would seek to use syntax in automatic MT evaluation as well. The first step in this direction was by (Liu and Gildea, 2005), who used syntactic structure and dependency information in order to see past the surface phenomena. Two of these metrics are based on matching syntactic subtrees between the translation and the reference, and the third is based on matching headword chains, but only for *unlabelled* dependencies. Since then, (Owczarzak et al., 2007a; Owczarzak et al., 2007b) have extended this line of research with the use of a term-based encoding of LFG *labelled* dependency graphs into unordered sets of dependency triples, and calculating precision, recall, and f-measure on the sets corresponding to the translation and reference sentences. With the addition of partial matching and *n*-best parses, (Owczarzak et al., 2007a; Owczarzak et al., 2007b) considerably outperform Liu and Gildea's (2005) highest correlations with human judgement.

Another line of research has led to machine learning methods which learn directly from human judgements (Ye et al., 2007). In this paper, we combine the syntax (dependency)-based and the machine learning-based approaches, and show greater correlations with human judgement than (Owczarzak et al., 2007a; Owczarzak et al., 2007b). We use both Ranking and Regression Support Vector Machines (SVMs) (Burges, 1998) in a range of experiments on different language pairs and data sets. We also examine the extent to which our novel method can be generalised across differ-

Proceedings of the 13th Annual Conference of the EAMT, pages 44–51,
Barcelona, May 2009

ent tasks and domains.

The remainder of the paper is organised as follows. In section 2, we outline approaches to automatic MT evaluation which are relevant to our work. In particular, in section 3 we describe the LFG labelled dependency approach of (Owczarzak et al., 2007a; Owczarzak et al., 2007b). In section 4, we demonstrate how labelled dependencies can be matched using SVMs, and describe the range of experiments carried out in section 5. The paper ends with our concluding remarks together with avenues for further research.

2 Evaluation Metrics in MT

Automatic evaluation metrics enable researchers to validate and optimise translation methods quickly. Simple *n*-gram-based metrics such as BLEU (Papineni et al., 2002) are fundamental to the development and tuning of MT systems. However, these *n*-gram-based metrics suffer from several shortcomings, such as low correlation with human judgement on the sentence level, exhibiting a bias towards statistical systems (Callison-Burch et al., 2006), and inconsistency in related evaluation scenarios (Chiang et al., 2008).

Many approaches have been taken to overcome the insufficiencies of BLEU. Word-based metrics like METEOR (Banerjee and Lavie, 2005) try to improve on the matching scheme; paraphrase-based methods such as ParaEval incorporate paraphrases extracted from an external data source (Zhou et al., 2006); syntactic methods try to use syntax information in hypothesis and reference (cf. section 2.1); and machine learning methods learn directly from human judgements (cf. section 2.2).

2.1 Dependency-based Metrics

The shortcomings of *n*-gram metrics have led a number of researchers to exploit more grammatical information in the hypothesis and reference sentences.

Syntactic features were first introduced in MT evaluation in (Liu and Gildea, 2005), who developed several metrics using constituency or dependency structure. (Owczarzak et al., 2007a; Owczarzak et al., 2007b) improved on the dependency matching of (Liu and Gildea, 2005) by using *n*-best labelled dependency triples produced by an LFG parser, so that parser noise is reduced and partial matchings can be found. (Kahn et al., 2008) match *n*-best head-modifier dependencies extracted from *n*-best constituency parses. They also consider the probabilities given by the constituency parser.

Dependency information is also used in metrics that incorporate different information sources. (Giménez and Màrquez, 2008) experimented using different levels of linguistic features and dependency relation-based metrics are among their best metrics at both system and sentence levels. Machine learning metrics such as (Ye et al., 2007) and (Albrecht and Hwa, 2007) also use some head-modifier dependency matches or dependency chains as features.

2.2 Machine Learning-based Metrics

Three kinds of machine learning-based approaches have been used in MT evaluation: (i) *Classification*-based approaches (Corston-Oliver et al., 2001) train a classifier to discriminate between the reference and the hypothesis. The higher the likelihood of a hypothesis being a reference, the better its quality is assumed to be; (ii) *Regression*-based methods (Albrecht and Hwa, 2007) train a model to try to reproduce the human judgement scores for each translation hypothesis; (iii) *Ranking*-based approaches (Ye et al., 2007) train a model with the ranking of different hypotheses on a particular sentence instead of the values of the scores.

Among these three approaches, classification only captures the difference between the hypotheses and the reference but ignores any differences in quality among these hypotheses. Both ranking- and regression-based methods have been reported to be successful in various MT evaluation tasks. In our experiments we combine them with the dependency-based method of (Owczarzak et al., 2007a; Owczarzak et al., 2007b) and directly compare them in a ranking task.

3 LFG Labelled Dependencies

Our work extends the method of (Owczarzak et al., 2007a; Owczarzak et al., 2007b) who use labelled dependencies in Lexical-Function Grammar (LFG). In LFG, a sentence is represented in both a hierarchical tree structure (C-structure) which captures the organisation of a sentence, and a set of labelled dependencies (F-structure). The dependencies in LFG are attribute-value features such as `subj(arrive, Julie)` or `pers(Julie,`

3) which capture the grammatical relations between constituents. They are more precise than head-modifier unlabelled dependencies. Here the trigram (DEP, HEAD, MODIFIER) is called a triple. In subj(arrive, Julie), DEP is subj, HEAD is arrive and MODIFIER is Julie.

In (Owczarzak et al., 2007a; Owczarzak et al., 2007b) it is shown that LFG F-structures can capture variations between sentences. For example, "Julie arrived yesterday." and "Yesterday Julie arrived." have only one bigram (Julie, arrived) in common but the same F-structures. This feature can help us better judge how similar a reference sentence and a hypothesis sentence are in MT evaluation.

3.1 Matching of Dependency Triples

To utilise LFG dependencies in MT evaluation, we use the LFG parser described in (Cahill et al., 2004) to generate dependency triples and perform matching on the triples. A hypothesis sentence is considered of higher quality when it has more triples matched with the reference sentence.

We perform three kinds of dependency matchings in our experiment: exact matching, partial matching, and WordNet extended matching. In exact matching all three elements in the triple must be the same to complete a match. With respect to the previous example, in partial matching, two triples can have different HEAD or MODIFIER values, whereas in WordNet extended matching, HEAD and MODIFIER can be substituted by synonyms in WordNet.

We only perform partial and WordNet extended matching on PREDICATE-ONLY dependencies (Owczarzak et al., 2007a; Owczarzak et al., 2007b). Both exact and partial matches on dependency type x are counted as one match on type x. A WordNet extended match is counted as one match on type x_WN.

3.2 Parser Noise and Matching in *n*-best Parses

The outputs of MT systems are often syntactically ill-formed and this makes it difficult for parsers to generate plausible parses. To compensate for this problem, we parse the hypothesis and reference translations to obtain the 50-best parses of each. Using the 50-best parses increases the chance of finding the correct match between the hypotheses and references.

For each pair of parses, we match the dependency triples, and select the pair of parses that has the highest F-score (cf. (3)) as matching and output the matching detail of this pair. Details on the effect of multiple parses can be found in (Owczarzak et al., 2007a; Owczarzak et al., 2007b).

3.3 Calculation of Matching Percentage

There are two ways of normalising the number of matchings. We can normalise with respect to the total number of triples in the hypothesis sentence (precision matching), as in (1):

$$P = \frac{\#matching\_triples}{\#triples\_in\_hypothesis} \quad (1)$$

or the total number of triples in the reference sentence (recall matching), as in (2):

$$R = \frac{\#matching\_triples}{\#triples\_in\_reference} \quad (2)$$

In (Owczarzak et al., 2007a; Owczarzak et al., 2007b), precision matching and recall matching are combined into an F-score, as in (3):

$$F = \frac{2PR}{P+R} \quad (3)$$

When using this combination, the relative weights of precision and recall are implicitly set to 1:1. In our experiment this combination is not necessary, as we can use both precision and recall values as features and let the SVM determine the respective weights of precision and recall.

4 Combining Labelled Dependency Matches with SVM

4.1 SVM in MT Evaluation

We use Ranking and Regression Support Vector Machines (Burges, 1998) in our experiments. Both Ranking and Regression SVMs assign a score to an input instance z, as in (4):

$$f(z) = \sum_{i=1}^{m} \alpha_i y_i \Phi(x_i) \cdot \Phi(z) + b \quad (4)$$

where (x_i, y_i) is the training example and Φ is the transformation function which transforms the input space to the feature space. However, the quantitative value from a ranking SVM is meaningless and only indicates its ranking.

The output of a ranking SVM aims at producing the correct rank of input examples, whereas regression SVMs aim at producing a value corresponding

to the input. Thus the ranking SVM maximises τ on a training set, where r_i is the metric ranking of systems on sentence i and r_i^* is the human ranking on sentence i, as in (5):

$$\frac{1}{n}\sum_{i=1}^{n}\tau(r_i, r_i^*) \quad (5)$$

Note that Kendall's τ measures the relevance of two rankings: $\tau(r_a, r_b) = \frac{P-Q}{P+Q}$, where P and Q are the amount of concordant and discordant pairs in r_a and r_b.

Regression SVMs, by contrast, are directly modelled on the human judgement scores by minimising (6):

$$\frac{1}{2}\sum_{i=1}^{m}(y_i - f(x_i))^2 \quad (6)$$

4.2 Kernels of SVM

We can often find a kernel function K in (4) with $K(x, z) = \Phi(x) \cdot \Phi(z)$. Kernel functions implicitly transform the input space into the feature space, while computation is still done in the input space.

We use three kinds of kernels in our experiments: (i) *Linear* kernels, the simplest form of kernel which do not transform the input space:

$$K(x, z) = x \cdot z \quad (7)$$

(ii) *Polynomial* kernels:

$$K(x, z) = (\alpha + \beta x \cdot z)^p \quad (8)$$

In our experiments α and β are set to 1, and p is set to 3.

(iii) *Radial Basis Function (RBF)* kernels:

$$K(x, z) = exp(-\gamma||x - z||^2) \quad (9)$$

The RBF kernel is the most complex kernel of the three. In some NLP tasks such as text categorisation (Joachims, 1998), RBF kernels are shown to capture the characteristics of the training data more accurately than linear or polynomial kernels. In our experiments γ is set to 1.

4.3 Normalisation of Features

The features in our experiments are the matching percentages on different types of dependencies. We propose two ways of normalising the value: horizontal and vertical. In horizontal normalisation, the number of matches on a certain dependency type are normalised by the total number of triples in the test/reference (based on whether precision or recall dependency matching is used) sentence, as in (10):

$$H(i) = \frac{\#matching\_depType(i)}{\#allTypes} \quad (10)$$

In vertical normalisation, only the number of dependencies of the same type are considered, as in (11):

$$V(i) = \frac{\#matching\_depType(i)}{\#depType(i)} \quad (11)$$

In horizontal normalisation, dependency types x and x_WN are counted separately. However, in vertical normalisation x_WN is counted as x, as x_WN is produced during matching, and we do not have this dependency type in the test or reference sentences.

Our horizontal normalisation is equivalent to the approach of (Ye et al., 2007). The vertical normalisation is a more radical approach to reflect the relative ratio of matches on different dependency types.

5 Experiments

5.1 Data

We use two data sets in our experiments. We use the WMT08 evaluation shared task dataset for Ranking SVM training and testing. We use 3,249 human rankings on outputs from different MT systems. The rankings are just a reflection of the relative quality of these systems; no absolute scores are given. We use 177 sentences from the Czech–English News Commentary task and 123 sentences from the Czech–English News task as our development set (DEV). We use 358 Czech–English News task sentences as the test set (TEST).

For the regression SVM we use the MTC4 corpus from LDC. The corpus consists of human-assigned fluency and adequacy scores to 11,028 outputs of MT systems. We remove the outputs that cause parser errors, leaving 11,004 segments, of which 2,000 sentences are used as our DEV set, 2,004 are used as the TEST set and the remaining 7,000 are used for training.

For generalisablity testing we also run experiments on WMT08 data with regression models generated from MTC4 data, and we run cross-language and cross-domain tests on WMT08 data.

Table 1: Ranking SVM: Different Kernels. Cons.: Consistency percentage; Corr.: Spearman's coefficient

	Cons. DEV	Corr. DEV	Cons. TEST	Corr. TEST
BLEU-4	0.3397	0.1896	0.2515	0.1297
BLEU-4s	0.5251	0.0909	0.5480	0.1427
LFG-F	0.5892	0.2521	0.5565	0.1796
PR-HV-L	**0.6325**	**0.2753**	**0.6055**	**0.2057**
PR-HV-P	0.6083	0.2548	0.5202	0.0806
PR-HV-R	0.5667	0.2008	0.5117	-0.0006

Table 2: Ranking SVM: Different Data Representation

	Cons. DEV	Corr. DEV	Cons. TEST	Corr. TEST
BLEU-4	0.3397	0.1896	0.2515	0.1297
BLEU-4s	0.5251	0.0909	0.5480	0.1427
LFG-F	0.5892	0.2521	0.5565	0.1796
P-V-L	0.5632	0.1599	0.5373	0.1227
P-H-L	0.5407	0.1315	0.5437	0.1565
R-V-L	0.6152	**0.2815**	0.5287	0.1656
R-H-L	0.5771	0.1988	0.5309	0.1903
PR-V-L	0.6170	0.2686	0.5884	**0.2206**
PR-H-L	0.6048	0.2178	**0.6055**	0.1939
P-HV-L	0.5685	0.1764	0.5415	0.1039
R-HV-L	0.6153	0.2751	0.5522	0.2068
PR-HV-L	**0.6325**	0.2753	**0.6055**	0.2057

5.2 Experimental Settings

We tested the ranking SVM with different types of feature representation. Normalisation (Norm) is performed with the horizontal (H), vertical (V) or both (HV) methods. Dependency matching (DEP) is computed in terms of precision (P), recall (R) or both (PR). We test with SVMs of linear (L), polynomial (P) and RBF (R) kernels (KERNEL) using the SVMLight software. Each configuration is denoted with {NORM}-{DEP}-{KERNEL} in both ranking and regression experimental results.

We use the following three metrics as baselines: BLEU (BLEU-4), add-one BLEU (BLEU-4s) and the labelled LFG-based metric (LFG-F) as described in (Owczarzak et al., 2007a; Owczarzak et al., 2007b). Note that the result of the LFG-F metric would have among the highest correlations with human judgement in the WMT08 shared evaluation task.

5.3 Experiments on Ranking SVM

We train ranking SVMs on WMT08 data to produce rankings of different system outputs on the same sentence.

Usually, the correlation between a metric and human rankings can be measured by Spearman's rank order correlation, defined in (12), where d is the difference between corresponding values in rankings and n is the length of the rankings:

$$\rho = 1 - (\frac{6 \sum d^2}{n(n^2-1)}) \quad (12)$$

However, in (Callison-Burch et al., 2008), it is argued that averaging ρ is meaningless, and so pair-wise consistent percentage is used instead to measure correlations in the WMT08 shared evaluation task. The pair-wise consistent percentage is equal to the number of correct pair-wise comparisons made by a metric divided by the total number of pair-wise comparisons to make.

We report both consistent percentage and sentence-level Spearman's correlation in our experiments. The Spearman's correlation is first computed on each ranking, and then averaged.

We explore the choice of different {KERNEL}s (Table 1) with PR-HV data representation (the best representation) and the choice of different {NORM}alization and {DEP}endency matching schemes (Table 2) with linear kernel (the best kernel).

In our experiments, PR-HV-L, the metric that uses all variations of features, yields the best overall results and outperforms the baseline on both DEV and TEST sets. A number of observations present themselves: (i) In Table 2, recall-based dependency match rates appear to be better features than precision-based rates. This pattern is also observed in other metrics such as METEOR. This is another example of the importance of recall in MT evaluation; (ii) In Table 1, more sophisticated kernels such as Polynomial and RBF kernels do not increase the performance of the metric and sometimes even decrease it. This might appear surprising, yet recall that we reserved all Czech–English translations for the development and test sets, so the SVM is not exposed to any human judgements on this language pair during training. We did this in order to show the generality of our machine learning-based method, but in so doing we may have caused the more sophisticated kernels to overfit on other language pairs. It tells us that selection of features is more important for our method than the learning algorithm itself; (iii) Vertical match features produce some good results but are more prone to overfitting. Using the RBF kernel on vertical match features often leads to lower correlations. The problem with the vertical match feature is that it ignores the total number of dependencies

Table 3: Regression SVM: Different Kernels. F/A Corr.: Correlation on fluency/adequacy

	F Corr. DEV	A Corr. DEV	F Corr. TEST	A Corr. TEST
BLEU-4	0.0679	0.145	0.1179	0.2087
BLEU-4s	0.0919	0.2077	0.1724	0.2499
LFG-F	**0.1076**	0.2926	0.2453	0.3779
R-H-L	0.0812	0.2987	**0.2506**	**0.3992**
R-H-P	0.0869	**0.2998**	0.2322	0.3948
R-H-R	0.0880	0.2996	0.2302	0.3935

Table 4: Regression SVM: Different Data Representation

	F Corr. DEV	A Corr. DEV	F Corr. TEST	A Corr. TEST
BLEU-4	0.0679	0.145	0.1179	0.2087
BLEU-4s	0.0919	0.2077	0.1724	0.2499
LFG-F	**0.1076**	0.2926	0.2453	0.3779
P-V-L	0.0961	0.2025	0.1993	0.2723
P-H-L	0.1030	0.2331	0.2040	0.2723
R-V-L	0.0694	0.2698	0.2222	0.3894
R-H-L	0.0812	0.2987	**0.2506**	0.3992
PR-V-L	0.0793	0.2669	0.2189	0.3827
PR-H-L	0.0989	**0.3027**	0.2436	0.3934
P-HV-L	0.1040	0.2165	0.2112	0.2894
R-HV-L	0.0850	0.2867	0.2307	**0.3999**
PR-HV-L	0.0933	0.2828	0.2288	0.3911

in a sentence. As a result, an output that correctly translates `subj` in a simple sentence with 2 dependencies will receive the same score as an output that only translates `subj` correctly in a compound sentence of 20 dependencies. This leads to problematic features and the problem might be exacerbated during learning; and (iv) When H, V, P and R are all used as features, we obtain the best overall result. This suggests that our different methods of normalisation and dependency matching are complementary in our ranking experiment.

5.4 Experiments on Regression SVM

In the regression SVM experiment, we use SVM to learn the scores which are assigned by human judges. The models for predicting fluency and adequacy scores are trained separately.

We calculate Pearson's correlation on both fluency and adequacy. Pearson's correlation is defined as:

$$r = \frac{1}{n-1} \sum (\frac{x_i - \bar{X}}{s_X})(\frac{y_i - \bar{Y}}{s_Y}) \qquad (13)$$

where x_i is the value of the i^{th} score, $\bar{X}$ is the mean score and s_X is the standard deviation.

The results on different kernels and different data representation are reported in Table 3 and Table 4 respectively. For the regression task, we test the choice of kernels on the R-V representation, which performs better than PR-HV in this task. In this experiment, we do not see a particular metric that consistently outperforms the baseline with respect to fluency. However, all metrics that are based on horizontal normalisation and recall-style dependency matching perform better than the baseline with respect to adequacy, for several reasons. Firstly, the features of our SVM models are the decomposed parts of LFG-F. LFG-F is better at evaluating adequacy than fluency (Owczarzak et al., 2007a; Owczarzak et al., 2007b). Thus we have better features for our adequacy-predicting SVM model.

Secondly, note that the fluency correlation on the DEV set is generally at a very low level, which indicates that the sentences in our DEV set are very hard to judge with respect to fluency. At this level, many trivial reasons can lead to an increase or decrease in correlation. In general, we can consider our R-H-L and PR-H-L metrics to be on a par with the baseline as far as fluency is concerned.

Except for the variance in fluency and adequacy, many tendencies observed in our ranking experiment still apply here. The recall-based features still prevail and sophisticated kernels do not improve performance. Vertical normalisation has a bigger negative impact in this experiment. It suggests that regression is more error-prone than ranking, perhaps because regression is harder.

5.5 Cross-Task Generalisability

We choose the two best-performing (R-H-L, PR-H-L) as well as two somewhat mediocre (R-HV-L, PR-HV-L) regression models and use them to compute scores for our ranking DEV and TEST set. We do not run this experiment in the opposite direction, because the MTC4 data is not collected in a ranking scenario and we consider it incomparable to the results on WMT08. We calculate Spearman's coefficient between the rankings induced from these regression scores and the human rankings to validate the generalisability of our learning method. For regression SVMs trained on MTC4, WMT08 is a corpus that is different with respect to language pair, domain, and evaluation criterion. The results are shown in Table 5.

Basically all four metrics trained on MTC4 outperform the LFG F-Score baseline on the TEST set, but are on a par or inferior on the DEV set. We consider this tendency to be related to the differ-

Table 5: Cross-Task Experiments

	Cons. DEV	Corr. DEV	Cons. TEST	Corr. TEST
BLEU-4	0.3397	0.1896	0.2515	0.1297
BLEU-4s	0.5251	0.0909	0.5480	0.1427
LFG-F	**0.5892**	**0.2521**	0.5565	0.1796
R-H	0.5875	0.2269	0.5714	**0.2471**
PR-H	0.5823	0.2152	**0.5991**	0.2084
R-HV	0.5649	0.1526	0.5479	0.1931
PR-HV	0.5719	0.1700	0.5714	0.1638

Table 6: Cross-Language Pair Experiments

	French		Other	
	Cons.	Corr.	Cons.	Corr.
BLEU	0.2795	0.1913	0.3652	0.1827
BLEU-4s	0.5818	0.2255	0.5675	0.1980
LFG-F	0.6204	0.2550	**0.5994**	**0.2503**
PR-V-L	0.6159	0.2420	0.5813	0.1848
PR-H-L	**0.6522**	**0.3131**	0.5844	0.2118
PR-HV-L	0.6227	0.2706	0.5896	0.1931

ence in domains. The ranking DEV set is dominated by commentary data, but the TEST set consists of news data only, which is identical to the MTC4 corpus we use to train the Regression SVM.

The results show that our method is generalisable to different tasks and evaluation criteria. When tested on similar domains, our regression SVM not only performs better than a very high baseline, but also approaches the performance of the best SVM trained specially for Ranking. Furthermore, the better performing metrics on MTC4 continue to perform well on WMT08.

However, our method is quite sensitive to domain change. The regression SVM trained on a completely different domain performs worse than the Ranking SVM on the DEV set, whereas on the TEST set it performs better than the Ranking SVM, which is trained on a multi-domain corpus.

5.6 Cross-Language Pair and Cross-Domain Generalisability

We carried out more experiments on the WMT08 data to explore the generalisability of our method over different language pairs and different domains. As far as language pair generalisability is concerned, we divide the dataset by language pairs into French–English and Other–English parts. We train the metrics on half of the French–English data, and test the model on the other half as well as Other–English data. The results are provided in Table 6.

For domain generalisability, we train the metrics on half of the News data and test them on the other half, as well as non-News data. The results are shown in Table 7. In both experiments we test with three metrics: PR-V-L, PR-H-L and PR-HV-L.

Table 7: Cross-Domain Experiments

	News		Non-News	
	Cons.	Corr.	Cons.	Corr.
BLEU	0.3035	0.1653	0.4739	0.2906
BLEU-4s	0.5548	0.2013	0.6277	0.2992
LFG-F	0.6112	0.2905	**0.6313**	**0.3007**
PR-V-L	0.6102	0.2540	0.5858	0.2088
PR-H-L	**0.6208**	**0.2957**	0.6129	0.2745
PR-HV-L	0.6134	0.2694	0.5996	0.2285

In both tests our methods do not outperform the baseline on different language pairs or domains. This is because our training set is very small. We are actually using a model trained on just hundreds of samples to rank thousands of samples in a different language pair/domain. In this context, all the tested methods obtain consistent percentages very close to the baseline in the cross-language pair experiment. It confirms that our method is more generalisable over different language pairs, and is somewhat more sensitive to changes in domains.

The shortcomings of vertical normalisation are magnified in these experiments. The correlations of our metrics on out-of-domain test sets follows the pattern of H $>$ HV $>$ V, which indicates that vertical normalisation causes performance to deteriorate. It accords with our assumption in the regression experiment that vertical normalisation is more prone to error on harder tasks.

6 Conclusion and Further Work

In this paper, we have presented a novel approach to automatic MT evaluation, where the labelled dependency approach of (Owczarzak et al., 2007a; Owczarzak et al., 2007b) is combined with the use of both Ranking and Regression Support Vector Machines (SVMs) (Burges, 1998). In our approach, we learn the required labelled dependencies, and show that our method improves over the approach of (Owczarzak et al., 2007a; Owczarzak et al., 2007b) with respect to correlation with human judgements. In addition, we demonstrate that our method is generalisable over different language pairs, but is somewhat more sensitive to changes in domains.

As far as extensions to this work are concerned, we aim to experiment with more features to improve cross-domain adaptability and to prevent any

overfitting. In addition, a more in-depth analysis needs to be carried out in order to discover which particular features contribute most to the correlation with human judgement.

Acknowledgements

We are grateful to Science Foundation Ireland (`http://www.sfi.ie`) grant number 07/CE/I1142 for sponsoring this research.

References

Albrecht, Joshua and Rebecca Hwa. 2007. Regression for sentence-level mt evaluation with pseudo references. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 296–303, Prague, Czech Republic.

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

Burges, Christopher J. C. 1998. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, 2(2):121–167.

Cahill, Aoife, Michael Burke, Ruth O'Donovan, Josef Van Genabith, and Andy Way. 2004. Long-distance dependency resolution in automatically acquired wide-coverage PCFG-based LFG approximations. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 319–326, Barcelona, Spain.

Callison-Burch, Chris, Miles Osborne, and Philipp Koehn. 2006. Re-evaluation the role of BLEU in machine translation research. In *EACL-2006, 11th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings*, pages 249–256, Trento, Italy.

Callison-Burch, Chris, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2008. Further meta-evaluation of machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 70–106, Columbus, OH.

Chiang, David, Steve DeNeefe, Yee Seng Chan, and Hwee Tou Ng. 2008. Decomposability of translation metrics for improved evaluation and efficient algorithms. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 610–619, Honolulu, HI.

Chiang, David. 2005. A Hierarchical Phrase-Based Model for Statistical Machine Translation. In *43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, MI.

Corston-Oliver, Simon, Michael Gamon, and Chris Brockett. 2001. A machine learning approach to the automatic evaluation of machine translation. In *Proceedings of 39th Annual Meeting and 10th Meeting of the European Chapter of the Association for Computational Linguistics*, pages 148–155, Toulouse, France.

Giménez, Jesús and Lluís Màrquez. 2008. A smorgasbord of features for automatic MT evaluation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 195–198, Columbus, OH.

Joachims, Thorsten. 1998. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, pages 137 – 142, Berlin/Heidelberg. Springer.

Kahn, Jeremy G., Mari Ostendorf, and Brian Roark. 2008. Automatic syntactic MT evaluation with expected dependency pair match. In *Proceedings of the Workshop Metrics MATR - Metrics for Machine Translation Challenge, Eighth Conference of the Association for Machine Translation in the Americas*, Waikiki, HI.

Liu, Ding and Daniel Gildea. 2005. Syntactic features for evaluation of machine translation. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 25–32, Ann Arbor, MI.

Owczarzak, Karolina, Josef van Genabith, and Andy Way. 2007a. Evaluating Machine Translation with LFG Dependencies. *Machine Translation*, 21(2):95–119.

Owczarzak, Karolina, Josef van Genabith, and Andy Way. 2007b. Labelled dependencies in machine translation evaluation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 104–111, Prague, Czech Republic.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA.

Ye, Yang, Ming Zhou, and Chin-Yew Lin. 2007. Sentence level machine translation evaluation as a ranking. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 240–247, Prague, Czech Republic.

Zhou, Liang, Chin-Yew Lin, Dragos Stefan Munteanu, and Eduard Hovy. 2006. Paraeval: Using paraphrases to evaluate summaries automatically. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 447–454, New York City, NY.