The DCU Dependency-Based Metric in WMT-MetricsMATR 2010

Yifan He Jinhua Du Andy Way Josef van Genabith
Centre for Next Generation Localisation
School of Computing
Dublin City University
Dublin 9, Ireland
{yhe,jdu,away,josef}@computing.dcu.ie

Abstract

We describe DCU's LFG dependency-based metric submitted to the shared evaluation task of WMT-MetricsMATR 2010.

The metric is built on the LFG F-structure-based approach presented in (Owczarzak et al., 2007). We explore the following improvements on the original metric: 1) we replace the in-house LFG parser with an open source dependency parser that directly parses strings into LFG dependencies; 2) we add a stemming module and unigram paraphrases to strengthen the aligner; 3) we introduce a chunk penalty following the practice of METEOR to reward continuous matches; and 4) we introduce and tune parameters to maximize the correlation with human judgement. Experiments show that these enhancements improve the dependency-based metric's correlation with human judgement.

1 Introduction

String-based automatic evaluation metrics such as BLEU (Papineni et al., 2002) have led directly to quality improvements in machine translation (MT). These metrics provide an alternative to expensive human evaluations, and enable tuning of MT systems based on automatic evaluation results.

However, there is widespread recognition in the MT community that string-based metrics are not discriminative enough to reflect the translation quality of today's MT systems, many of which have gone beyond pure string-based approaches (cf. (Callison-Burch et al., 2006)).

With that in mind, a number of researchers have come up with metrics which incorporate more sophisticated and linguistically motivated resources. Examples include METEOR (Banerjee and Lavie, 2005; Lavie and Denkowski, 2009) and TERP (Snover et al., 2010), both of which now utilize stemming, WordNet and paraphrase information. Experimental and evaluation campaign results have shown that these metrics can obtain better correlation with human judgements than metrics that only use surface-level information.

Given that many of today's MT systems incorporate some kind of syntactic information, it was perhaps natural to use syntax in automatic MT evaluation as well. This direction was first explored by (Liu and Gildea, 2005), who used syntactic structure and dependency information to go beyond the surface level matching.

Owczarzak et al. (2007) extended this line of research with the use of a term-based encoding of Lexical Functional Grammar (LFG:(Kaplan and Bresnan, 1982)) *labelled* dependency graphs into unordered sets of dependency triples, and calculating precision, recall, and F-score on the triple sets corresponding to the translation and reference sentences. With the addition of partial matching and n-best parses, Owczarzak et al. (2007)'s method considerably outperforms Liu and Gildea's (2005) w.r.t. correlation with human judgement.

The EDPM metric (Kahn et al., 2010) improves this line of research by using arc labels derived from a Probabilistic Context-Free Grammar (PCFG) parse to replace the LFG labels, showing that a PCFG parser is sufficient for preprocessing, compared to a dependency parser in (Liu and Gildea, 2005) and (Owczarzak et al., 2007). EDPM also incorporates more information sources: e.g. the parser confidence, the Porter stemmer, WordNet synonyms and paraphrases.

Besides the metrics that rely solely on the dependency structures, information from the dependency parser is a component of some other metrics that use more diverse resources, such as the textual entailment-based metric of (Pado et al., 2009).

In this paper we extend the work of (Owczarzak et al., 2007) in a different manner: we use an

Proceedings of the Joint 5th Workshop on Statistical Machine Translation and MetricsMATR, pages 349–353,
Uppsala, Sweden, 11-16 July 2010.

adapted version of the Malt parser (Nivre et al., 2006) to produce 1-best LFG dependencies and allow triple matches where the dependency labels are different. We incorporate stemming, synonym and paraphrase information as in (Kahn et al., 2010), and at the same time introduce a chunk penalty in the spirit of METEOR to penalize discontinuous matches. We sort the matches according to the match level and the dependency type, and weight the matches to maximize correlation with human judgement.

The remainder of the paper is organized as follows. Section 2 reviews the dependency-based metric. Sections 3, 4, 5 and 6 introduce our improvements on this metric. We report experimental results in Section 7 and conclude in Section 8.

2 The Dependency-Based Metric

In this section, we briefly review the metric presented in (Owczarzak et al., 2007).

2.1 C-Structure and F-Structure in LFG

In Lexical Functional Grammar (Kaplan and Bresnan, 1982), a sentence is represented as both a hierarchical c-(onstituent) structure which captures the phrasal organization of a sentence, and a f-(unctional) structure which captures the functional relations between different parts of the sentence. Our metric currently only relies on the f-structure, which is encoded as labeled dependencies in our metric.

2.2 MT Evaluation as Dependency Triple Matching

The basic method of (Owczarzak et al., 2007) can be illustrated by the example in Table 1.

The metric in (Owczarzak et al., 2007) performs triple matching over the Hyp- and Ref-Triples and calculates the metric score using the F-score of matching precision and recall. Let m be the number of matches, h be the number of triples in the hypothesis and e be the number of triples in the reference. Then we have the matching precision $P = m/h$ and recall $R = m/e$. The score of the hypothesis in (Owczarzak et al., 2007) is the F-score based on the precision and recall of matching as in (1):

$$Fscore = \frac{2PR}{P+R} \quad (1)$$

Table 1: Sample Hypothesis and Reference

Hypothesis
rice will be held talks in egypt next week
Hyp-Triples
`adjunct(will, rice)`
`xcomp(will, be)`
`adjunct(talks, held)`
`xcomp(be, talks)`
`adjunct(talks, in)`
`obj(in, egypt)`
`adjunct(week, next)`
`adjunct(talks, week)`
Reference
rice to hold talks in egypt next week
Ref-Triples
`obl(rice, to)`
`obj(hold, to)`
`adjunct(week, talks)`
`adjunct(talks, in)`
`obj(in, egypt)`
`adjunct(week, next)`
`obj(hold, week)`

2.3 Details of the Matching Strategy

(Owczarzak et al., 2007) uses several techniques to facilitate triple matching. First of all, considering that the MT-generated hypotheses have variable quality and are sometimes ungrammatical, the metric will search the 50-best parses of both the hypothesis and reference and use the pair that has the highest F-score to compensate for parser noise.

Secondly, the metric performs *complete* or *partial* matching according to the dependency labels, so the metric will find more matches on dependency structures that are presumably more informative.

More specifically, for all except the LFG `Predicate-Only` labeled triples of the form `dep(head, modifier)`, the method does not allow a match if the dependency labels (`deps`) are different, thus enforcing a *complete* match. For the `Predicate-Only` dependencies, *partial* matching is allowed: i.e. two triples are considered identical even if only the `head` or the `modifier` are the same.

Finally, the metric also uses linguistic resources for better coverage. Besides using WordNet synonyms, the method also uses the lemmatized output of the LFG parser, which is equivalent to using

an English lemmatizer.

If we do not consider these additional linguistic resources, the metric would find the following matches in the example in Table 1: `adjunct(talks, in)`, `obj(in, egypt)` and `adjunct(week, next)`, as these three triples appear both in the reference and in the hypothesis.

2.4 Points for Improvement

We see several points for improvement from Table 1 and the analysis above.

- More linguistic resources: we can use more linguistic resources than WordNet in pursuit of better coverage.
- Using the 1-best parse instead of 50-best parses: the parsing model we currently use does not produce k-best parses and using only the 1-best parse significantly improves the speed of triple matching. We allow 'soft' triple matches to capture the triple matches which we might otherwise miss using the 1-best parse.
- Rewarding continuous matches: it would be more desirable to reflect the fact that the 3 matching triples `adjunct(talks, in)`, `obj(in, egypt)` and `adjunct(week, next)` are continuous in Table 1.

We introduce our improvements to the metric in response to these observations in the following sections.

3 Producing and Matching LFG Dependency Triples

3.1 The LFG Parser

The metric described in (Owczarzak et al., 2007) uses the DCU LFG parser (Cahill et al., 2004) to produce LFG dependency triples. The parser uses a Penn treebank-trained parser to produce c-structures (constituency trees) and an LFG f-structure annotation algorithm on the c-structure to obtain f-structures. In (Owczarzak et al., 2007), triple matching on f-structures produced by this paradigm correlates well with human judgement, but this paradigm is not adequate for the WMT-MetricsMatr evaluation in two respects: 1) the in-house LFG annotation algorithm is not publicly available and 2) the speed of this paradigm is not satisfactory.

We instead use the Malt Parser[1] (Nivre et al., 2006) with a parsing model trained on LFG dependencies to produce the f-structure triples. Our collaborators[2] first apply the LFG annotation algorithm to the Penn Treebank training data to obtain f-structures, and then the f-structures are converted into dependency trees in CoNLL format to train the parsing model. We use the *liblinear* (Fan et al., 2008) classification module to for fast parsing speed.

3.2 Hard and Soft Dependency Matching

Currently our parser produces only the 1-best outputs. Compared to the 50-best parses in (Owczarzak et al., 2007), the 1-best parse limits the number of triple matches that can be found. To compensate for this, we allow triple matches that have the same `Head` and `Modifier` to constitute a match, even if their dependency labels are different. Therefore for triples `Dep1(Head1, Mod1)` and `Dep2(Head2, Mod2)`, we allow three types of match: a *complete* match if the two triples are identical, a *partial* match if `Dep1=Dep2` and `Head1=Head2`, and a *soft* match if `Head1=Head2` and `Mod1=Mod2`.

4 Capturing Variations in Language

In (Owczarzak et al., 2007), lexical variations at the word-level are captured by WordNet. We use a Porter stemmer and a unigram paraphrase database to allow more lexical variations.

With these two resources combined, there are four stages of word level matching in our system: *exact* match, *stem* match, *WordNet* match and unigram *paraphrase* match. The stemming module uses Porter's stemmer implementation[3] and the WordNet module uses the JAWS WordNet interface.[4] Our metric only considers unigram paraphrases, which are extracted from the paraphrase database in TERP[5] using the script in the METEOR[6] metric.

[1] `http://maltparser.org/index.html`

[2] Özlem Çetinoğlu and Jennifer Foster at the National Centre for Language Technology, Dublin City University

[3] `http://tartarus.org/~martin/PorterStemmer/`

[4] `http://lyle.smu.edu/~tspell/jaws/index.html`

[5] `http://www.umiacs.umd.edu/~snover/terp/`

[6] `http://www.cs.cmu.edu/~alavie/METEOR/`

5 Adding Chunk Penalty to the Dependency-Based Metric

The metric described in (Owczarzak et al., 2007) does not explicitly consider word order and fluency. METEOR, on the other hand, utilizes this information through a chunk penalty. We introduce a chunk penalty to our dependency-based metric following METEOR's string-based approach.

Given a reference $r = w_{r1}...w_{rn}$, we denote w_{ri} as 'covered' if it is the head or modifier of a matched triple. We only consider the w_{ri}s that appear as `head` or `modifier` in the reference triples. After this notation, we follow METEOR's approach by counting the number of chunks in the reference string, where a chunk $w_{rj}...w_{rk}$ is a sequence of adjacent covered words in the reference. Using the hypothesis and reference in Table 1 as an example, the three matched triples `adjunct(talks, in)`, `obj(in, egypt)` and `adjunct(week, next)` will *cover* a continuous word sequence in the reference (underlined), constituting one single chunk:

rice to hold talks (in) egypt next week

Based on this observation, we introduce a similar chunk penalty Pen as in METEOR in our metric, as in 2:

$$Pen = \gamma \cdot (\frac{\#chunks}{\#matches})^{\beta} \quad (2)$$

where β and γ are free parameters, which we tune in Section 6.2. We add this penalty to the dependency based metric (cf. Eq. (1)), as in Eq. (3).

$$score = (1 - Pen) \cdot Fscore \quad (3)$$

6 Parameter Tuning

6.1 Parameters of the Metric

In our metric, dependency triple matches can be categorized according to many criteria. We assume that some matches are more critical than others and encode the importance of matches by weighting them differently. The final match will be the sum of weighted matches, as in (4):

$$m = \sum \lambda_t m_t \quad (4)$$

where λ_t and m_t are the weight and number of match category t. We categorize a triple match according to three perspectives: 1) the level of match L={*complete, partial*}; 2) the linguistic resource used in matching R={*exact, stem, WordNet, paraphrase*}; and 3) the type of dependency D. To avoid too large a number of parameters, we only allow a set of frequent dependency types, along with the type *other*, which represents all the other types and the type *soft* for *soft* matches. We have D={*app, subj, obj, poss, adjunct, topicrel, other, soft*}.

Therefore for each triple match m, we can have the type of the match $t \in L \times R \times D$.

6.2 Tuning

In sum, we have the following parameters to tune in our metric: precision weight α, chunk penalty parameters β, γ, and the match type weights $\lambda_1...\lambda_n$. We perform Powell's line search (Press et al., 2007) on the sufficient statistics of our metric to find the set of parameters that maximizes Pearson's ρ on the segment level. We perform the optimization on the MT06 portion of the NIST MetricsMATR 2010 development set with 2-fold cross validation.

7 Experiments

We experiment with four settings of the metric: HARD, SOFT, SOFTALL and WEIGHTED in order to validate our enhancements. The first two settings compare the effect of allowing/not allowing *soft* matches, but only uses WordNet as in (Owczarzak et al., 2007). The third setting applies our additional linguistic features and the final setting tunes parameter weights for higher correlation with human judgement.

We report Pearson's r, Spearman's ρ and Kendall's τ on segment and system levels on the NIST MetricsMATR 2010 development set using Snover's scoring tool.[7]

Table 2: Correlation on the Segment Level

	r	ρ	τ
HARD	0.557	0.586	0.176
SOFT	0.600	0.634	0.213
SOFTALL	0.633	0.662	0.235
WEIGHTED	0.673	0.709	0.277

Table 2 shows that allowing *soft* triple matches and using more linguistic features all lead to higher correlation with human judgement. Though the parameters might somehow overfit on

[7] `http://www.umiacs.umd.edu/~snover/terp/scoring/`

the data set even if we apply cross validation, this certainly confirms the necessity of weighing dependency matches according to their types.

Table 3: Correlation on the System Level

	r	ρ	τ
HARD	0.948	0.905	0.786
SOFT	0.964	0.905	0.786
SOFTALL	0.975	0.976	0.929
WEIGHTED	0.989	1.000	1.000

When considering the system-level correlation in Table 3, the trend is very similar to that of the segment level. The improvements we introduce all lead to improvements in correlation with human judgement.

8 Conclusions and Future Work

In this paper we describe DCU's dependency-based MT evaluation metric submitted to WMT-MetricsMATR 2010. Building upon the LFG-based metric described in (Owczarzak et al., 2007), we use a publicly available parser instead of an in-house parser to produce dependency labels, so that the metric can run on a third party machine. We improve the metric by allowing more lexical variations and weighting dependency triple matches depending on their importance according to correlation with human judgement.

For future work, we hope to apply this method to languages other than English, and perform more refinement on dependency type labels and linguistic resources.

Acknowledgements

This research is supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Dublin City University. We thank Özlem Çetinoğlu and Jennifer Foster for providing us with the LFG parsing model for the Malt Parser, as well as the anonymous reviewers for their insightful comments.

References

Satanjeev Banerjee 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.

Aoife Cahill, 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-2004)*, pages 319–326, Barcelona, Spain.

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

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Jeremy G. Kahn, Matthew Snover, and Mari Ostendorf. 2010. Expected dependency pair match: predicting translation quality with expected syntactic structure. *Machine Translation*.

Ronald M. Kaplan and Joan Bresnan. 1982. Lexical-functional grammar: A formal system for grammatical representation. *The mental representation of grammatical relations*, pages 173–281.

Alon Lavie and Michael J. Denkowski. 2009. he meteor metric for automatic evaluation of machine translation. *Machine Translation*, 23(2-3).

Ding Liu 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.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. Maltparser: A data-driven parser-generator for dependency parsing. In *In The fifth international conference on Language Resources and Evaluation (LREC-2006)*, pages 2216–2219, Genoa, Italy.

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

Sebastian Pado, Michel Galley, Dan Jurafsky, and Christopher D. Manning. 2009. Robust machine translation evaluation with entailment features. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 297–305, Suntec, Singapore.

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

William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. 2007. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY.

Matthew Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. 2010. Ter-plus: paraphrase, semantic, and alignment enhancements to translation edit rate. *Machine Translation*.