PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 96 OCTOBER 2011 109–117

Optimising Multiple Metrics with MERT

Christophe Servan, Holger Schwenk

LIUM, University of Le Mans

Abstract

Optimisation in statistical machine translation is usually made toward the BLEU score, but this metric is questioned about its relevance to an human evaluation. Many other metrics exist but none of them are in perfect harmony with human evaluation. On the other hand, most evaluation campaigns use multiple metrics (BLEU, TER, METEOR, etc.). Statistical machine translation systems can be optimised for other metrics than BLEU, but usually the optimisation with other metrics tends to decrease the BLEU score, the main metric used in MT evaluation campaigns.

In this paper we extend the minimum error training tool of the popular Moses SMT toolkit with a scorer for the TER score, and any linear combination of the existing metrics. The TER scorer was reimplemented in C++ which results in a ten times faster execution than the reference java code.

We have performed experiments with two large-scale phrase-base SMT systems to show the benefit of the new options of the minimum error training in Moses. The first one translates from French into English (WMT 2011 evaluation). The second one was developed in the frame work of the DARPA Gale project to translate from Arabic to English in three different genres (news, web and transcribed broadcast news and conversations).

 Corresponding author: servan@lium.univ-lemans.fr
Cite as: Christophe Servan, Holger Schwenk. Optimising Multiple Metrics with MERT. The Prague Bulletin of Mathematical Linguistics No. 96, 2011, pp. 109–117. doi: 10.2478/v10108-011-0016-z.

1. Introduction

It is today common practice to use a log-linear approach to combine the various models involved in statistical machine translation (SMT). This is summarised in the fundamental equation of SMT:

$$e^* = \arg\max_e \log \prod_i f_i(e,f)^{\lambda_i} = \arg\max_e \sum_i \lambda_i \log f_i(e,f) \quad (1)$$

$f_i(e,f)$ are functions of the source e and target word f sequences. Typical feature functions include the translation and distortion model, a language model on the target language and various penalties. Each feature function is weighted by a coefficient λ_i. These weights are usually optimised so that to maximise the translation performance on some development data. In the popular Moses toolkit (Koehn et al., 2007), this numerically optimisation is performed by a tool called `mert` (Bertoldi et al., 2009) which performs a simplex-style optimisation.

The provided mert tool uses the BLEU score as performance measure of the translation quality (Papineni et al., 2002). Despite the fact that the relevance of BLEU is often questioned, see for instance (Hammon, 2007), it is still a metric frequently used to evaluate machine translation, and more importantly to tune SMT systems. In fact, many metrics have been proposed to measure MT quality, for example TER (Snover et al., 2006), TERp (Snover et al., 2009) and METEOR (Banerjee and Lavie, 2005) just to mention some, and many of them are believed to correlate better with human judgements of translation quality. However, many of these metrics are not used to tune the SMT systems, at least they are not available for the Moses toolkit. It has also been observed several times that it is better to optimise towards the same metric that is later used to evaluate the SMT system.

Some previous works showed the interest of optimising toward BLEU and TER (Mauser et al., 2008; Cer et al., 2010b). Some of this work has been done with other MT systems like the Phrasal Machine Translation system (Cer et al., 2010a). Most of people use the Moses SMT system (Koehn et al., 2007) which uses MERT to optimise its parameters. We implemented a metric combination into MERT.

This kind of experiment is hard to reproduce by the fact that TER and metric combination is not directly implemented in MERT program (Bertoldi et al., 2009). That's why a MT Marathon project was suggested on this subject last year. We start over this project and we provide this tool to the Machine Translation community as open source.

In this paper we describe an extension of the mert optimiser provided in the Moses toolkit to optimise the translations performance with respect to the linear combination of multiple metrics. We have performed experiments for two well known large translations tasks: a French/English SMT system that was ranked among the best ones in the 2011 WMT evaluation and a state-of-the-art SMT system to translate from Arabic to English in the framework of the DARPA Gale project. As as special case we

will consider the frequently used combination $(\mathrm{TER} - \mathrm{BLEU})/2$, but we also report results on other combinations.

This paper is organised as follows. In the next section we will first give some details on the new scorers for the mert tool. We then present experimental results for the two tasks mentioned above. The paper concludes with a discussion of open issues.

2. A fast scorer for TER

We extended the mert tool in a flexible way so that to allow multiple metrics. The mert tool provided with Moses uses the notion of a `scorer`. This is basically an abstract C++ class that implements a particular metric, by default either the BLEU, WER and PER scores. It is not possible to use a combination of several metrics. Each scorer reads a file with n-best translations and produces a file with the corresponding scores.

We realised two additional scorers:

- the *TER scorer* which implements the translation edit rate (Snover et al., 2006) algorithm in MERT;
- the *merge scorer* which implements the combination of two or more metrics.

The TER score is usually calculated using the reference implementation of M. Snover in java. We reimplemented a TER scorer in C++ in order to have an easier interface with the mert software and to speed up the calculation of the TER score. In fact, it was observed that the calculation of the TER score with the java software can take some time, up to a minute on large development sets. This would result in a slow optimisation by MERT since the scorer is called on large n-best lists. Our implementation in C++ is roughly ten times faster than the java code.

In addition, we implemented a `merge` scorer that allows the linear combination of an arbitrary number of scorers. This allows in particular to minimise $(\mathrm{TER}-\mathrm{BLEU})/2$, but it also possible to attach a weight to each scorer, for instance $(\mathrm{TER} - 2 * \mathrm{BLEU})/2$. For this, a new switch has been added to the script *mert-moses.pl* `--sc-config`, *e.g.* `--sc-config=BLEU:2,TER:1`. All necessary configuration files are generated automatically by the script *mert-moses.pl*. A typical configuration file is shown in Table 1. This configuration file is used by the `merge` scorer in order to set weights associated with metrics. When this scorer is used, the extractor software, which is a part of the mert toolkit, succesively extract data (features and scores) for each metric. Then, the mert software is called by using the switch `--sctype MERGE`.

The mert tool always tries to maximise the returned scores, but TER is an error metric that should be minimised. Therefore, the *negTER* score is used and actually returns 1-TER.

Theses scorers are open source and released to the machine translation community with the moses SMT toolkit.

Metric	weight	feature file name	score file name
BLEU	2	BLEU_FEATURE_FILE	BLEU_SCORING_FILE
TER	1	TER_FEATURE_FILE	TER_SCORING_FILE

Table 1. Example of a configuration file for the metric combination 2xBLEU-TER.

3. Experiments

The TER scorer as well as the merge scorer were evaluated for two important tasks: the translation from French to English and the translation from Arabic to English. Both system are phrase-based, but the same procedure could also be applied to the hierarchical system `moses_chart`. We performed several experiments to asses the impact of different combinations of the scorers, keeping all other settings unchanged, in particular a fixed seed was used during the mert optimisation process. The beam search is set to 0.4 and the mert n-best is set to 100.

3.1. French/English system

Since several years, between several European languages. LIUM build this a system to translate between French and English. Our official systems are optimised using the default implementation of the mert tool which only optimises toward the BLEU score. After the evaluation, we have performed additional experiments with our new scorer. In this paper we only consider the translation from French to English. In our experiments with different metrics, we used exactly the same translation and language models than in our evaluation systems. The first model was trained on about 435M words of parallel data, while more than 7 billion words were used for the English language model. More details are given in (Schwenk et al., 2011). We report BLEU and TER scores on our development corpus (*newstest2009*), our internal test set (*newstest2010*) and the official test set of this year's evaluation (*newstest2011*). All metrics are case sensitive and include punctuations.

Table 2 summarises all the results. The first line, labelled BLEU corresponds to our official evaluation system. The second line, shows the results when using negTER as optimisation metric instead of BLEU. It is not surprising to see that this leads a decrease in the TER score, but unfortunately this comes at the cost of a worse BLEU score. BLEU is a precision metric which must be maximised while TER is an error measure which should be minimised. Therefore, it is common practice to look simultaneously at both metrics using the value $(\mathrm{TER} - \mathrm{BLEU})/2$ which must be of course minimised.

It can be clearly seen that we achieve best results by directly optimising the combined score $(\mathrm{TER} - \mathrm{BLEU})/2$. On the development data, this decreases the TER score from 53.98 to 53.58 without penalising the BLEU score. This results in an improvement of the combined score from 12.42 to 12.22. It is nice to see that the results are even

	newstest2009 (Dev)			newstest2010 (Internal test)			newstest2011 (Evaluation test)		
Optimisation	BLEU	TER	$\frac{\text{TER}-\text{BLEU}}{2}$	BLEU	TER	$\frac{\text{TER}-\text{BLEU}}{2}$	BLEU	TER	$\frac{\text{TER}-\text{BLEU}}{2}$
BLEU	29.14	53.98	12.42	29.65	52.78	11.57	30.19	51.61	10.71
TER	27.65	52.91	12.63	28.79	51.56	11.39	29.36	50.57	10.61
1xBLEU-TER	29.15	53.58	**12.22**	29.95	52.42	**11.24**	30.37	51.36	**10.50**
2xBLEU-TER	29.10	53.88	12.39	29.93	52.55	11.31	30.15	51.56	10.71
3xBLEU-TER	29.19	53.83	12.32	29.99	52.46	**11.24**	30.14	51.56	10.71
4xBLEU-TER	29.21	54.01	12.40	29.98	52.60	11.31	30.08	51.75	10.84
5xBLEU-TER	29.33	53.84	12.26	29.89	52.53	11.32	30.21	51.56	10.68

Table 2. Results for the French/English WMT 2011 translation task.

better on the internal and official test set: the BLEU and the TER score do improve when optimising on the combined criterion instead of BLEU itself. On *newstest2011* BLEU improves from 30.19 to 30.37 and TER from 51.61 to 51.36. Unfortunately, this improved system did not participate in the human evaluation. It would be very interesting to see how theses changes impact human judgements.

The merge scorer is able to perform arbitrary linear combinations of the two metrics. The corresponding results are shown in the subsequent lines of Table 2. For this task, this did not improve the overall combined performance.

3.2. GALE evaluation task

In 2005 DARPA lunched a new 5 year language technology program called Global Autonomous Language Exploitation, shortly GALE. The goal of this project was to build high performance machine translation systems from Arabic and Mandarin into English for text and speech and to prepare this information in various ways (*distillation*). Several genres were considered: news paper texts, WEB data and broadcast news and conversations automatically transcribed from speech to text. LIUM developed phrase-based systems to translate all these genres from Arabic to English in collaboration with IBM's Rosetta team.

DARPA organised yearly evaluations to measure the progress. The official metric of this evaluations was HTER which is a human judgement. In principle, this error measure corresponds to the minimal number of edit operations (insertion, deletion, substitution and block shift) a human operator has to perform to correct the errors of the automatic translation. Obviously, the human metric HTER is related to the automatic metric TER. In fact, HTER could be seen as TER with optimal references created on the fly for each sentence, or TER with respect to a pool of all possible reference translations.

We developed separate systems for the news, web and speech genre.[1] Statistics on the used parallel training data are given in Table 3. For each genre a development

[1] it is not possible to automatically separate broadcast news and broadcast conversations.

and internal test corpus was available. It consisted of about 50k English words for the news and web genre, and almost 100k words for the speech genre. Three reference translations are available for the web and broadcast conversation genres, while only one is available for the news and broadcast news genres.

Genre	# lines	# words AR	# words EN
news	3M	72.8M	76.9M
web	2.2M	46.6M	48.3M
speech	2.4M	54.4M	57.3M

Table 3. Size of the different bitexts used for our Arabic/English Gale systems.

All the experimental results are summarised in Table 4. Again, we give the BLEU score, TER and the combination $(\mathrm{TER}-\mathrm{BLEU})/2$ when optimising the systems for the different criteria.

Corpus name	Optimisation	Dev			Test		
		BLEU	TER	$\frac{\mathrm{TER}-\mathrm{BLEU}}{2}$	BLEU	TER	$\frac{\mathrm{TER}-\mathrm{BLEU}}{2}$
news	BLEU	33.56	43.80	5.12	33.56	44.25	5.34
	TER	34.07	42.81	**4.37**	34.07	43.18	**4.55**
	1xBLEU-TER	33.55	43.67	5.06	33.55	44.00	5.22
	2xBLEU-TER	33.47	43.66	5.09	33.47	44.05	5.29
	3xBLEU-TER	33.66	43.45	4.89	33.66	43.91	5.12
	4xBLEU-TER	33.63	43.68	5.03	33.63	44.01	5.19
	5xBLEU-TER	33.47	43.69	5.11	33.47	44.15	5.34
web	BLEU	40.78	61.20	10.96	39.27	61.86	11.29
	TER	40.46	60.59	**10.68**	39.24	61.43	**11.10**
	1xBLEU-TER	40.76	61.09	10.79	39.52	61.72	11.10
	2xBLEU-TER	40.62	61.01	10.87	39.28	61.56	11.14
	3xBLEU-TER	40.72	*60.86*	10.72	39.42	61.56	*11.07*
	4xBLEU-TER	40.71	61.17	10.92	39.33	61.69	11.18
	5xBLEU-TER	40.63	61.55	11.24	39.06	62.04	11.49
speech	BLEU	33.73	58.03	12.15	33.94	58.03	12.04
	TER	33.30	55.92	**11.31**	33.39	56.34	**11.47**
	1xBLEU-TER	34.04	56.98	11.47	34.13	57.17	11.52
	2xBLEU-TER	33.97	57.21	11.62	34.12	57.28	11.58
	3xBLEU-TER	33.86	57.97	12.05	33.88	58.13	12.12
	4xBLEU-TER	33.85	58.02	12.09	33.79	58.37	12.29
	5xBLEU-TER	33.85	57.91	12.03	33.84	58.13	12.14

Table 4. Results for the Arabic/English Gale translation tasks.

The results for the news genre are somehow surprising. In fact, when we tune on negTER we get, as expected, an improvement of the TER score on the development and test corpus, but the BLEU score also improves by about 0.5 points, in comparison to tuning directly on the BLEU score. Overall, the combined score $(\mathrm{TER} - \mathrm{BLEU})/2$ is substantially improved. Tuning directly on the combined metric always produced worse combined scores than tuning on negTER only. We are currently investigating this effect. Note that it can't be explained by the BLEU brevity penalty since it is 1.0 for all the experiments.

The improvements are less important for the web genre: we achieve a smaller improvement in the TER score, with only minor changes in the BLEU score. Optimising on TER or $(\mathrm{TER} - \mathrm{BLEU})/2$ gives basically the same results: 10.68 versus 10.72 on the dev data, and 11.10 and 11.07 on the test data. The improvements in the TER score for the speech genre are quite substantial, up to 2 points, with a modest loss in the BLEU score.

Overall, it is always best to tune on negTER for all the genres of the Arabic/English Gale systems, although $(\mathrm{TER}-\mathrm{BLEU})/2$ is almost quite as good for the web and speech genres.

4. Conclusion

This paper addressed the important issue on which automatic measure one should optimise the weights of the feature functions in the log-linear model used in SMT. For this, we extended the mert optimisation software in the very popular Moses SMT toolkit with scorers for TER and a *merge* scorer which allows to optimise an arbitrary linear combination of other metrics. Since the TER scorer is implemented in C++ in performs roughly ten times faster than the reference java code. The whole software is open-source and available in Moses svn[2].

We have performed experiments with two large-scale phrase-base SMT systems. The first one translates from French into English (WMT 2011 evaluation). The second one was developed in the frame work of the DARPA Gale project to translate from Arabic to English in three different genres (news, web and transcribed broadcast news and conversations). For the WMT system we have observed, like many others before, that tuning on one metric, concretely BLEU or TER, obviously improves the performance measured in this metric, but usually worsens other metric. Best results were obtained when tuning directly on a linear combination of both, usually $(\mathrm{TER} - \mathrm{BLEU})/2$.

For the Arabic/English system, significant improvements of the combined score $(\mathrm{TER} - \mathrm{BLEU})/2$ were obtained, in particular for the news and speech genre. However, in contrast to the WMT task, this can be obtained by tuning on TER only. We are currently investigating the reasons for these effects: is the tuning affected by the speci-

[2] `https://mosesdecoder.svn.sourceforge.net/svnroot/mosesdecoder`

ficities of the language pair, i.e. French/English versus Arabic/English, the number of available reference translations, ... ?

In the future, we plan to add further metrics, namely TERp and METEOR, and we try to study which metric combination is best related to human judgements.

5. Acknowledgements

This work was partially supported under the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-C-0022, and by the European Commission under the project EUROMATRIXPLUS.

Bibliography

Banerjee, Satanjeev and Alon Lavie. Meteor: An automatic metric for MT evaluation with improved correlation with human judgments. In *Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at the 43th Annual Meeting of the Association of Computational Linguistics*, 2005.

Bertoldi, Nicola, Barry Haddow, and Jean-Baptiste Fouet. Improved minimum error rate training in Moses. *The Prague Bulletin of Mathematical Linguistics*, 91, 2009.

Cer, Daniel, Michel Galley, Daniel Jurafsky, and Christopher Manning. Phrasal: A toolkit for statistical machine translation with facilities for extraction and incorporation of arbitrary model features. In *North American Association of Computational Linguistics - Demo Session (NAACL-10)*, 2010a.

Cer, Daniel, Christopher D. Manning, and Daniel Jurafsky. The best lexical metric for phrase-based statistical mt system optimization. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 555–563, Stroudsburg, PA, USA, 2010b. Association for Computational Linguistics. ISBN 1-932432-65-5. URL `http://portal.acm.org/citation.cfm?id=1857999.1858079`.

Hammon, Olivier. Rapport du projet CESTA : Campagne d'evaluation des systèmes de traduction automatique. Technical report, ELDA, 2007.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *ACL, demonstration session*, 2007.

Mauser, Arne, Saša Hasan, and Hermann Ney. Automatic evaluation measures for statistical machine translation system optimization. In *LREC'08*, 2008.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *ACL*, 2002.

Schwenk, Holger, Patrik Lambert, Loïc Barrault, Christophe Servan, Haithem Afli, Sadaf Abdul-Rauf, and Kashif Shah. LIUM's SMT machine translation systems for WMT 2011. In *6th Workshop on statistical Machine Translation*, 2011.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *ACL*, 2006.

Snover, Matthew, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. Fluency, adequacy, or HTER ? Exploring different human judgments with a tunable MT metric. In *Proceedings of the Fourth Workshop on Statistical Machine Translation at the 12th Meeting of the European Chapter of the Association for Computational Linguistics (EACL-2009)*, pages 259–268, 2009.

Address for correspondence:
Christophe Servan
servan@lium.univ-lemans.fr
LIUM, University of Le Mans
72085 Le Mans cedex 9, FRANCE