TESLA at WMT 2011: Translation Evaluation and Tunable Metric

Daniel Dahlmeier[1] and **Chang Liu**[2] and **Hwee Tou Ng**[1,2]
[1]NUS Graduate School for Integrative Sciences and Engineering
[2]Department of Computer Science, National University of Singapore
{danielhe,liuchan1,nght}@comp.nus.edu.sg

Abstract

This paper describes the submission from the National University of Singapore to the WMT 2011 Shared Evaluation Task and the Tunable Metric Task. Our entry is TESLA in three different configurations: TESLA-M, TESLA-F, and the new TESLA-B.

1 Introduction

TESLA (Translation Evaluation of Sentences with Linear-programming-based Analysis) was first proposed in Liu et al. (2010). The simplest variant, TESLA-M (M stands for *minimal*), is based on N-gram matching, and utilizes light-weight linguistic analysis including lemmatization, part-of-speech tagging, and WordNet synonym relations. TESLA-B (B stands for *basic*) additionally takes advantage of bilingual phrase tables to model phrase synonyms. It is a new configuration proposed in this paper. The most sophisticated configuration TESLA-F (F stands for *full*) additionally uses language models and a ranking support vector machine instead of simple averaging. TESLA-F was called TESLA in Liu et al. (2010). In this paper, we rationalize the naming convention by using TESLA to refer to the whole family of metrics.

The rest of this paper is organized as follows. Sections 2 to 4 describe the TESLA variants TESLA-M, TESLA-B, and TESLA-F, respectively. Section 5 describes MT tuning with TESLA. Section 6 shows experimental results for the evaluation and the tunable metric task. The last section concludes the paper.

2 TESLA-M

The version of TESLA-M used in WMT 2011 is exactly the same as in Liu et al. (2010). The description is reproduced here for completeness.

We consider the task of evaluating machine translation systems in the direction of translating a *source language* to a *target language*. We are given a *reference translation* produced by a professional human translator and a machine-produced *system translation*. At the highest level, TESLA-M is the *arithmetic average* of F-measures between *bags of N-grams* (BNGs). A BNG is a multiset of weighted N-grams. Mathematically, a BNG B consists of tuples (b_i, b_i^W), where each b_i is an N-gram and b_i^W is a positive real number representing the weight of b_i. In the simplest case, a BNG contains every N-gram in a translated sentence, and the weights are just the counts of the respective N-grams. However, to emphasize the content words over the function words, we discount the weight of an N-gram by a factor of 0.1 for every function word in the N-gram. We decide whether a word is a function word based on its POS tag.

In TESLA-M, the BNGs are extracted in the target language, so we call them *bags of target language N-grams* (BTNGs).

2.1 Similarity functions

To match two BNGs, we first need a similarity measure between N-grams. In this section, we define the similarity measures used in our experiments. We adopt the similarity measure from MaxSim (Chan and Ng, 2008; Chan and Ng, 2009) as s_{ms}. For unigrams x and y,

Proceedings of the 6th Workshop on Statistical Machine Translation, pages 78–84,
Edinburgh, Scotland, UK, July 30–31, 2011.

- If lemma(x) = lemma(y), then $s_{ms} = 1$.
- Otherwise, let

$$a = I(\text{synsets}(x) \text{ overlap with synsets}(y))$$
$$b = I(\text{POS}(x) = \text{POS}(y))$$

where $I(\cdot)$ is the indicator function, then $s_{ms} = (a + b)/2$.

The synsets are obtained by querying WordNet (Fellbaum, 1998). For languages other than English, a synonym dictionary is used instead.

We define two other similarity functions between unigrams:

$$s_{lem}(x, y) = I(\text{lemma}(x) = \text{lemma}(y))$$
$$s_{pos}(x, y) = I(\text{POS}(x) = \text{POS}(y))$$

All the three unigram similarity functions generalize to N-grams in the same way. For two N-grams $x = x^{1,2,...,n}$ and $y = y^{1,2,...,n}$,

$$s(x, y) = \begin{cases} 0 & \text{if } \exists i,\ s(x^i, y^i) = 0 \\ \frac{1}{n} \sum_{i=1}^{n} s(x^i, y^i) & \text{otherwise} \end{cases}$$

2.2 Matching two BNGs

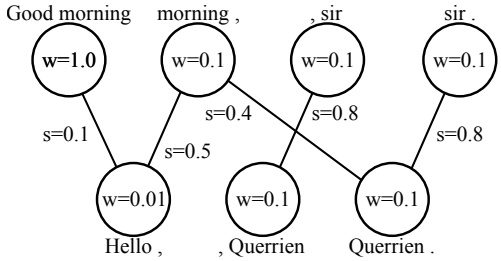
Now we describe the procedure of matching two BNGs. We take as input BNGs X and Y and a similarity measure s. The i-th entry in X is x_i and has weight x_i^W (analogously for y_j and y_j^W).

Intuitively, we wish to align the entries of the two BNGs in a way that maximizes the overall similarity. An example matching problem for bigrams is shown in Figure 1a, where the weight of each node is shown, along with the hypothetical similarity for each edge. Edges with a similarity of zero are not shown. Note that for each function word, we discount the weight by a factor of ten. The solution to the matching problem is shown in Figure 1b, and the overall similarity is $0.5 \times 0.01 + 0.8 \times 0.1 + 0.8 \times 0.1 = 0.165$.

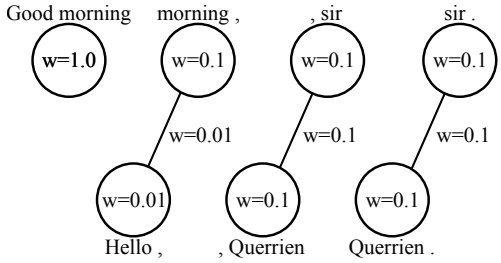
Mathematically, we formulate this as a (real-valued) linear programming problem[1]. The variables are the allocated weights for the edges

$$w(x_i, y_j) \quad \forall i, j$$

[1]While integer linear programming is NP-complete, real-valued linear programming can be solved efficiently.



(a) The matching problem



(b) The solution

Figure 1: A BNG matching problem

We maximize

$$\sum_{i,j} s(x_i, y_j) w(x_i, y_j)$$

subject to

$$\begin{aligned} w(x_i, y_j) &\geq 0 \quad \forall i, j \\ \sum_j w(x_i, y_j) &\leq x_i^W \quad \forall i \\ \sum_i w(x_i, y_j) &\leq y_j^W \quad \forall j \end{aligned}$$

The value of the objective function is the overall similarity S. Assuming X is the reference and Y is the system translation, we have

$$\text{Precision} = \frac{S}{\sum_j y_j^W}$$
$$\text{Recall} = \frac{S}{\sum_i x_i^W}$$

The F-measure is derived from the precision and the recall:

$$F = \frac{\text{Precision} \times \text{Recall}}{\alpha \times \text{Precision} + (1 - \alpha) \times \text{Recall}}$$

In this work, we set $\alpha = 0.8$, following MaxSim. The value gives more importance to the recall than the precision.

If the similarity function is binary-valued and transitive, such as s_{lem} and s_{pos}, then we can use a much simpler and faster greedy matching procedure: the best match is simply $\sum_g \min(\sum_{x_i=g} x_i^W, \sum_{y_i=g} y_i^W)$.

2.3 Scoring

The TESLA-M sentence-level score for a reference and a system translation is the arithmetic average of the BTNG F-measures for unigrams, bigrams, and trigrams based on similarity functions s_{ms} and s_{pos}. We thus have $3 \times 2 = 6$ BTNG F-measures for TESLA-M.

We can compute a system-level score for a machine translation system by averaging its sentence-level scores over the complete test set.

3 TESLA-B

TESLA-B uses the average of two types of F-measures: (1) BTNG F-measures as in TESLA-M and (2) F-measures between bags of N-grams in one or more pivot languages, called *bags of pivot language N-grams* (BPNGs), The rest of this section focuses on the *generation* of the BPNGs. Their matching is done in the same way as described for BTNGs in the previous section.

3.1 Phrase level semantic representation

Given a sentence-aligned bitext between the target language and a pivot language, we can align the text at the word level using well known tools such as GIZA++ (Och and Ney, 2003) or the Berkeley aligner (Liang et al., 2006; Haghighi et al., 2009).

We observe that the distribution of aligned phrases in a pivot language can serve as a semantic representation of a target language phrase. That is, if two target language phrases are often aligned to the same pivot language phrase, then they can be inferred to be similar in meaning. Similar observations have been made by previous researchers (Bannard and Callison-Burch, 2005; Callison-Burch et al., 2006; Snover et al., 2009).

We note here two differences from WordNet synonyms: (1) the relationship is not restricted to the word level only, and (2) the relationship is not binary. The degree of similarity can be measured by the percentage of overlap between the semantic representations.

3.2 Segmenting a sentence into phrases

To extend the concept of this semantic representation of phrases to sentences, we segment a sentence in the target language into phrases. Given a phrase table, we can approximate the probability of a phrase p by:

$$Pr(p) = \frac{N(p)}{\sum_{p'} N(p')} \quad (1)$$

where $N(\cdot)$ is the count of a phrase in the phrase table. We then define the likelihood of segmenting a sentence S into a sequence of phrases $(p_1, p_2, \ldots, p_n)$ by:

$$Pr(p_1, p_2, \ldots, p_n|S) = \frac{1}{Z(S)} \prod_{i=1}^{n} Pr(p_i) \quad (2)$$

where $Z(S)$ is a normalizing constant. The segmentation of S that maximizes the probability can be determined efficiently using a dynamic programming algorithm. The formula has a strong preference for longer phrases, as every $Pr(p)$ is a small fraction. To deal with out-of-vocabulary (OOV) words, we allow any single word w to be considered a phrase, and if $N(w) = 0$, we set $N(w) = 0.5$ instead.

3.3 BPNGs as sentence level semantic representation

Simply merging the phrase-level semantic representation is insufficient to produce a sensible sentence-level semantic representation. As an example, we consider two target language (English) sentences segmented as follows:

1. ||| *Hello ,* ||| *Querrien* ||| . |||
2. ||| *Good morning , sir .* |||

A naive comparison of the bags of aligned pivot language (French) phrases would likely conclude that the two sentences are completely unrelated, as the bags of aligned phrases are likely to be completely disjoint. We tackle this problem by constructing a confusion network representation of the aligned phrases, as shown in Figures 2 and 3. A confusion network is a compact representation of a potentially exponentially large number of weighted and likely malformed French sentences. We can collect the N-gram statistics of this ensemble of French sentences

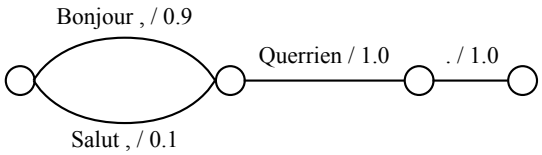


Figure 2: A confusion network as a semantic representation

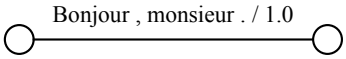


Figure 3: A degenerate confusion network as a semantic representation

efficiently from the confusion network representation. For example, the trigram *Bonjour , Querrien* [2] would receive a weight of $0.9 \times 1.0 = 0.9$ in Figure 2. As with BTNGs, we discount the weight of an N-gram by a factor of 0.1 for every function word in the N-gram, so as to place more emphasis on the content words.

The collection of all such N-grams and their corresponding weights forms the BPNG of a sentence. The reference and system BPNGs are then matched using the algorithm outlined in Section 2.2.

3.4 Scoring

The TESLA-B sentence-level score is a linear combination of (1) BTNG F-measures for unigrams, bigrams, and trigrams based on similarity functions s_{ms} and s_{pos}, and (2) BPNG F-measures for unigrams, bigrams, and trigrams based on similarity functions s_{lem} and s_{pos}. We thus have 3×2 F-measures from the BTNGs and $3 \times 2 \times$ *#pivot languages* F-measures from the BPNGs. We average the BTNG and BPNG scores to obtain s_{BTNG} and s_{BPNG}, respectively. The sentence-level TESLA-B score is then defined as $\frac{1}{2}(s_{\text{BTNG}} + s_{\text{BPNG}})$. The two-step averaging process prevents the BPNG scores from overwhelming the BTNG scores, especially when we have many pivot languages. The system-level TESLA-B score is the arithmetic average of the sentence-level TESLA-B scores.

[2] Note that an N-gram can span more than one segment.

4 TESLA-F

Unlike the simple arithmetic averages used in TESLA-M and TESLA-B, TESLA-F uses a general linear combination of three types of scores: (1) BTNG F-measures as in TESLA-M and TESLA-B, (2) BPNG F-measures as in TESLA-B, and (3) normalized language model scores of the system translation, defined as $\frac{1}{n} \log P$, where n is the length of the translation, and P the language model probability. The method of training the linear model depends on the development data. In the case of WMT, the development data is in the form of manual rankings, so we train SVM^{rank} (Joachims, 2006) on these instances to build the linear model. In other scenarios, some form of regression can be more appropriate.

The BTNG and BPNG scores are the same as used in TESLA-B. In the WMT campaigns, we use two language models, one generated from the Europarl dataset and one from the news-train dataset. We thus have 3×2 features from the BTNGs, $3 \times 2 \times$ *#pivot languages* features from the BPNGs, and 2 features from the language models. Again, we can compute system-level scores by averaging the sentence-level scores.

4.1 Scaling of TESLA-F Scores

While machine translation evaluation is concerned only with the relative order of the different translations but not with the absolute scores, there are practical advantages in normalizing the evaluation scores to a range between 0 and 1. For TESLA-M and TESLA-B, this is already the case, since every F-measure has a range of $[0, 1]$ and so do their averages. In contrast, the SVM^{rank}-produced model typically gives scores very close to zero.

To remedy that, we note that we have the freedom to scale and shift the linear SVM model without changing the metric. We observe that the F-measures have a range of $[0, 1]$, and studying the data reveals that $[-15, 0]$ is a good approximation of the range for normalized language model scores, for all languages involved in the WMT campaign. Since we know the range of values of an F-measure feature (between 0 and 1) and assuming that the range of the normalized LM score is between –15 and 0, we can find the maximum and minimum possible score given the weights. Then we linearly scale the range

of scores from [min, max] to [0, 1]. We provide an option of scaling TESLA-F scores in the new release of TESLA.

5 MT tuning with TESLA

All variants of TESLA can be used for automatic MT tuning using Z-MERT (Zaidan, 2009). Z-MERT's modular design makes it easy to integrate a new metric. As TESLA already computes scores at the sentence level, integrating TESLA into Z-MERT was straightforward. First, we created a "streaming" version of each TESLA metric which reads translation candidates from standard input and prints the sentence-level scores to standard output. This allows Z-MERT to easily query the metric for sentence-level scores during MT tuning. Second, we wrote a Java wrapper that calls the TESLA code from Z-MERT. The resulting metric can be used for MERT tuning in the standard fashion. All that a user has to do is to change the metric in the Z-MERT configuration file to TESLA. All the necessary code for Z-MERT tuning is included in the new release of TESLA.

6 Experiments

6.1 Evaluation Task

We evaluate TESLA using the publicly available data from WMT 2009 for into-English and out-of-English translation. The pivot language phrase tables and language models are built using the WMT 2009 training data. The SVM^{rank} model for TESLA-F is trained on manual rankings from WMT 2008. The results for TESLA-M and TESLA-F have previously been reported in Liu et al. (2010)[3]. We add results for the new variant TESLA-B here.

Tables 1 and 2 show the sentence-level consistency and system-level Spearman's rank correlation, respectively for into-English translation. For comparison, we include results for some of the best performing metrics in WMT 2009. Tables 3 and 4 show the same results for out-of-English translation. We do not include the English-Czech language pair in our experiments, as we unfortunately do not have good linguistic resources for the Czech language.

[3]The English-Spanish system correlation differs from our previous result after fixing a minor mistake in the language model.

	cz-en	fr-en	de-en	es-en	hu-en	Overall
TESLA-M	0.60	0.61	0.61	0.59	0.63	0.61
TESLA-B	0.63	0.64	0.63	0.62	0.63	0.63
TESLA-F	0.63	0.65	0.64	0.62	0.66	0.63
ulc	0.63	0.64	0.64	0.61	0.60	0.63
maxsim	0.60	0.63	0.63	0.61	0.62	0.62
meteor-0.6	0.47	0.51	0.52	0.49	0.48	0.50

Table 1: Into-English sentence-level consistency on WMT 2009 data

	cz-en	fr-en	de-en	es-en	hu-en	Avg
TESLA-M	1.00	0.86	0.85	0.99	0.66	0.87
TESLA-B	1.00	0.92	0.67	0.95	0.83	0.87
TESLA-F	1.00	0.92	0.68	0.94	0.94	0.90
ulc	1.00	0.92	0.78	0.86	0.60	0.83
maxsim	0.70	0.91	0.76	0.98	0.66	0.80
meteor-0.6	0.70	0.93	0.56	0.87	0.54	0.72

Table 2: Into-English system-level Spearman's rank correlation on WMT 2009 data

The new TESLA-B metric proves to be competitive to its siblings and is often on par with the more sophisticated TESLA-F metric. The exception is the English-German language pair, where TESLA-B has very low system-level correlation. We have two possible explanations for this. First, the system-level correlation is computed on a very small sample size (the ranked list of MT systems). This makes the system-level correlation score more volatile compared to the sentence-level consistency score which is computed on thousands of sentence pairs. Second, German has a relatively free word order which potentially makes word alignment and phrase table extraction more noisy. Interestingly, all participating metrics in WMT 2009 had low system-level correlation for the English-German language pair.

	en-fr	en-de	en-es	Overall
TESLA-M	0.64	0.59	0.59	0.60
TESLA-B	0.65	0.59	0.60	0.61
TESLA-F	0.68	0.57	0.60	0.61
wpF	0.66	0.60	0.61	0.61
wpbleu	0.60	0.47	0.49	0.51

Table 3: Out-of-English sentence-level consistency on WMT 2009 data

	en-fr	en-de	en-es	Avg
TESLA-M	0.93	0.86	0.79	0.86
TESLA-B	0.91	0.05	0.63	0.53
TESLA-F	0.85	0.78	0.67	0.77
wpF	0.90	-0.06	0.58	0.47
wpbleu	0.92	0.07	0.63	0.54

Table 4: Out-of-English system-level Spearman's rank correlation on WMT 2009 data

Tune\Test	BLEU	TESLA-M	TESLA-B	TESLA-F
BLEU	**0.2715**	0.3756	0.3129	0.3920
TESLA-M	0.2279	0.4056	0.3279	0.3981
TESLA-B	0.2370	0.4001	0.3257	0.3977
TESLA-F	0.2432	**0.4076**	**0.3299**	**0.4007**

Table 5: Automatic evaluation scores on held out test portion for the tunable metric task. The best result in each column is printed in bold.

6.2 Tunable Metric Task

The goal of the new tunable metric task is to explore MT tuning with metrics other than BLEU (Papineni et al., 2002). To allow for a fair comparison, the WMT organizers provided participants with a complete Joshua MT system for an Urdu-English translation task. We tuned models for each variant of TESLA, using Z-MERT in the default configuration provided by the organizers. There are four reference translations for each Urdu source sentence. The size of the N-best list is set to 300.

For our own experiments, we randomly split the development set into a development portion (781 sentences) and a held-out test portion (200 sentences). We run the same Z-MERT tuning process for each TESLA variant on this reduced development set and evaluate the resulting models on the held out test set. We include a model trained with BLEU as an additional reference point. The results are shown in Table 5. We observe that the model trained with TESLA-F achieves the best results when evaluated with any of the TESLA metrics, although the differences between the scores are small. We found that TESLA produces slightly longer translations than BLEU: 22.4 words (TESLA-M), 21.7 words (TESLA-B), and 22.5 words (TESLA-F), versus 18.7 words (BLEU). The average reference length is 19.8 words.

The official evaluation for the tunable metric task is performed using manual rankings. The score of a system is calculated as the percentage of times the system is judged to be either better or equal (*score1*) or strictly better (*score2*) compared to each other system in pairwise comparisons. Although we submit results for all TESLA variants, only our primary submission TESLA-F is included in the manual evaluation. The results for TESLA-F are mixed. When evaluated with score1, TESLA-F is ranked 7th out of 8 participating systems, but when evaluated with score2, TESLA-F is ranked second best. These findings differ from previous results that we reported in Liu et al. (2011) where MT systems tuned with TESLA-M and TESLA-F consistently outperform two other systems tuned with BLEU and TER for translations from French, German, and Spanish into English on the WMT 2010 news data set. A manual inspection of the references in the tunable metric task shows that the translations are of lower quality compared to the news data sets used in WMT. As the SVM model in TESLA-F is trained with rankings from WMT 2008, it is possible that the model is less robust when applied to Urdu-English translations. This could explain the mixed performance of TESLA-F in the tunable metric task.

7 Conclusion

We introduce TESLA-B, a new variant of the TESLA machine translation metric and present experimental results for all TESLA variants in the setting of the WMT evaluation task and tunable metric task. All TESLA variants are integrated into Z-MERT for automatic machine translation tuning.

Acknowledgments

This research was done for CSIDM Project No. CSIDM-200804 partially funded by a grant from the National Research Foundation (NRF) administered by the Media Development Authority (MDA) of Singapore.

References

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*.

Chris Callison-Burch, Philipp Koehn, and Miles Osborne. 2006. Improved statistical machine translation using paraphrases. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

Yee Seng Chan and Hwee Tou Ng. 2008. MaxSim: A maximum similarity metric for machine translation evaluation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*.

Yee Seng Chan and Hwee Tou Ng. 2009. MaxSim: Performance and effects of translation fluency. *Machine Translation*, 23(2–3):157–168.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.

Aria Haghighi, John Blitzer, John DeNero, and Dan Klein. 2009. Better word alignments with supervised ITG models. In *Proceedings of 47th Annual Meeting of the Association for Computational Linguistics and the 4th IJCNLP of the AFNLP*.

Thorsten Joachims. 2006. Training linear SVMs in linear time. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining*.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

Chang Liu, Daniel Dahlmeier, and Hwee Tou Ng. 2010. TESLA: Translation evaluation of sentences with linear-programming-based analysis. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*.

Chang Liu, Daniel Dahlmeier, and Hwee Tou Ng. 2011. Better evaluation metrics lead to better machine translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1).

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

Matthew Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. 2009. Fluency, adequacy, or HTER? Exploring different human judgments with a tunable MT metric. In *Proceedings of of the Fourth Workshop on Statistical Machine Translation*.

Omar Zaidan. 2009. Z-MERT: A fully configurable open source tool for minimum error rate training of machine translation systems. *The Prague Bulletin of Mathematical Linguistics*, 91:79–88.