TESLA: Translation Evaluation of Sentences with Linear-programming-based Analysis

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

Abstract

We present TESLA-M and TESLA, two novel automatic machine translation evaluation metrics with state-of-the-art performances. TESLA-M builds on the success of METEOR and MaxSim, but employs a more expressive linear programming framework. TESLA further exploits parallel texts to build a shallow semantic representation. We evaluate both on the WMT 2009 shared evaluation task and show that they outperform all participating systems in most tasks.

1 Introduction

In recent years, many machine translation (MT) evaluation metrics have been proposed, exploiting varying amounts of linguistic resources.

Heavyweight linguistic approaches including RTE (Pado et al., 2009) and ULC (Giménez and Màrquez, 2008) performed the best in the WMT 2009 shared evaluation task. They exploit an extensive array of linguistic features such as parsing, semantic role labeling, textual entailment, and discourse representation, which may also limit their practical applications.

Lightweight linguistic approaches such as METEOR (Banerjee and Lavie, 2005), MaxSim (Chan and Ng, 2008), wpF and wpBleu (Popović and Ney, 2009) exploit a limited range of linguistic information that is relatively cheap to acquire and to compute, including lemmatization, part-of-speech (POS) tagging, and synonym dictionaries.

Non-linguistic approaches include BLEU (Papineni et al., 2002) and its variants, TER (Snover et al., 2006), among others. They operate purely at the surface word level and no linguistic resources are required. Although still very popular with MT researchers, they have generally shown inferior performances than the linguistic approaches.

We believe that the lightweight linguistic approaches are a good compromise given the current state of computational linguistics research and resources. In this paper, we devise TESLA-M and TESLA, two lightweight approaches to MT evaluation. Specifically: (1) the core features are F-measures derived by matching bags of N-grams; (2) both recall and precision are considered, with more emphasis on recall; and (3) WordNet synonyms feature prominently.

The main novelty of TESLA-M compared to METEOR and MaxSim is that we match the N-grams under a very expressive linear programming framework, which allows us to assign weights to the N-grams. This is in contrast to the greedy approach of METEOR, and the more restrictive maximum bipartite matching formulation of MaxSim.

In addition, we present a heavier version TESLA, which combines the features using a linear model trained on development data, making it easy to exploit features not on the same scale, and leaving open the possibility of domain adaptation. It also exploits parallel texts of the target language with other languages as a shallow semantic representation, which allows us to model phrase synonyms and idioms. In contrast, METEOR and MaxSim are capable of processing only word synonyms from WordNet.

The rest of this paper is organized as follows. Section 2 gives a high level overview of the evaluation task. Sections 3 and 4 describe TESLA-M and TESLA, respectively. Section 5 presents experimental results in the setting of the WMT 2009 shared evaluation task. Finally, Section 6 concludes the paper.

2 Overview

We consider the task of evaluating machine translation systems in the direction of translating the *source language* to the *target language*. Given a *reference translation* and a *system translation*, the

Proceedings of the Joint 5th Workshop on Statistical Machine Translation and MetricsMATR, pages 354–359,
Uppsala, Sweden, 15-16 July 2010.

goal of an automatic machine translation evaluation algorithm such as TESLA(-M) is to output a score predicting the quality of the system translation. Neither TESLA-M nor TESLA requires the source text, but as additional linguistic resources, TESLA makes use of phrase tables generated from parallel texts of the target language and other languages, which we refer to as *pivot languages*. The source language may or may not be one of the pivot languages.

3 TESLA-M

This section describes TESLA-M, the lighter one among the two metrics. 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 its weight. 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).

3.1 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 as s_{ms}. For unigrams x and y,

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

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

  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 $x = x^{1,2,\ldots,n}$ and $y = y^{1,2,\ldots,n}$,

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

3.2 Matching two BNGs

Now we describe the procedure of matching two BNGs. We take as input the following:

1. Two BNGs, X and Y. The ith entry in X is x_i and has weight x_i^W (analogously for y_j and y_j^W).
2. A similarity measure, s, that gives a similarity score between any two entries in the range of 0 to 1.

Intuitively, we wish to align the entries of the two BNGs in a way that maximizes the overall similarity. As translations often contain one-to-many or many-to-many alignments, we allow one entry to split its weight among multiple alignments. An example matching problem is shown in Figure 1a, where the weight of each node is shown, along with the similarity for each edge. Edges with a similarity of zero are not shown. The solution to the matching problem is shown in Figure 1b, and the overall similarity is $0.5 \times 1.0 + 0.5 \times 0.6 + 1.0 \times 0.2 + 1.0 \times 0.1 = 1.1$.

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

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

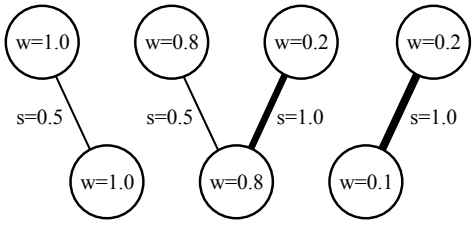
We maximize

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

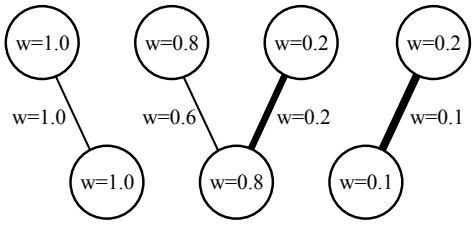
subject to

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

[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

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.

3.3 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 $3 \times 2 = 6$ features 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.

3.4 Reduction

When every x_i^W and y_j^W is 1, the linear programming problem proposed above reduces to *weighted bipartite matching*. This is a well known result; see for example, Cormen et al. (2001) for details. This is the formalism of MaxSim, which precludes the use of fractional weights.

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 $\sum_g \min(\sum_{x_i=g} x_i^W, \sum_{y_i=g} y_i^W)$.

4 TESLA

Unlike the simple arithmetic average used in TESLA-M, TESLA uses a general linear combination of three types of features: BTNG F-measures as in TESLA-M, F-measures between bags of N-grams in each of the pivot languages, called *bags of pivot language N-grams* (BPNGs), and 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 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.

4.1 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. For example, at the word level,

the phrases *good morning* and *hello* are unrelated even with a synonym dictionary, but they both very often align to the same French phrase *bonjour*, and we conclude they are semantically related to a high degree.

4.2 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.

4.3 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. ||| *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 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.

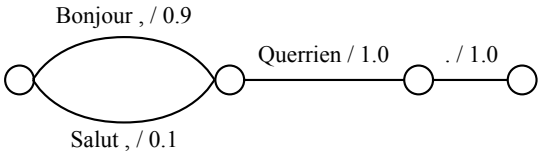


Figure 2: A confusion network as a semantic representation

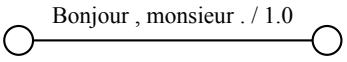


Figure 3: A degenerate confusion network as a semantic representation

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 3.2.

4.4 Scoring

The TESLA 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}, (2) BPNG F-measures for unigrams, bigrams, and trigrams based on similarity functions s_{lem} and s_{pos} for each pivot language, and (3) normalized language model scores. In this work, we use two language models. 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.

5 Experiments

5.1 Setup

We test our metrics in the setting of the WMT 2009 evaluation task (Callison-Burch et al., 2009). The manual judgments from WMT 2008 are used

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

as the development data and the metric is evaluated on WMT 2009 manual judgments with respect to two criteria: sentence level consistency and system level correlation.

The sentence level consistency is defined as the percentage of correctly predicted pairs among all the manually judged pairs. Pairs judged as ties by humans are excluded from the evaluation. The system level correlation is defined as the average Spearman's rank correlation coefficient across all translation tracks.

5.2 Pre-processing

We POS tag and lemmatize the texts using the following tools: for English, OpenNLP POS-tagger[3] and WordNet lemmatizer; for French and German, TreeTagger[4]; for Spanish, the FreeLing toolkit (Atserias et al., 2006); and for Czech, the Morce morphological tagger[5].

For German, we additionally perform noun compound splitting. For each noun, we choose the split that maximizes the geometric mean of the frequency counts of its parts, following the method in (Koehn and Knight, 2003):

$$\max_{n,p_1,p_2,\ldots,p_n} \left[\prod_{i=1}^{n} N(p_i)\right]^{\frac{1}{n}}$$

The resulting compound split sentence is then POS tagged and lemmatized.

Finally, we remove all non-alphanumeric tokens from the text in all languages. To generate the language model features, we train SRILM (Stolcke, 2002) trigram models with modified Kneser-Ney discounting on the supplied monolingual Europarl and news commentary texts.

We build phrase tables from the supplied news commentary bitexts. Word alignments are produced by the Berkeley aligner. The widely used phrase extraction heuristic in (Koehn et al., 2003) is used to extract phrase pairs and phrases of up to 4 words are collected.

5.3 Into-English task

For each of the BNG features, we generate three scores, for unigrams, bigrams, and trigrams respectively. For BPNGs, we generate one such triple for each of the four pivot languages supplied, namely Czech, French, German, and Spanish.

[3] opennlp.sourceforge.net

[4] www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger

[5] ufal.mff.cuni.cz/morce/index.php

	System correlation	Sentence consistency
TESLA	0.8993	0.6324
TESLA-M	0.8718	0.6097
ulc	0.83	0.63
maxsim	0.80	0.62
meteor-0.6	0.72	0.50

Table 1: Into-English task on WMT 2009 data

Table 1 compares the scores of TESLA and TESLA-M against three participants in WMT 2009 under identical settings[6]: ULC (a heavyweight linguistic approach with the best performance in WMT 2009), MaxSim, and METEOR. The results show that TESLA outperforms all these systems by a substantial margin, and TESLA-M is very competitive too.

5.4 Out-of-English task

A synonym dictionary is required for target languages other than English. We use the freely available Wiktionary dictionary[7] for each language. For Spanish, we additionally use the Spanish WordNet, a component of FreeLing.

Only one pivot language (English) is used for the BPNG. For the English-Czech task, we only have one language model instead of two, as the Europarl language model is not available.

Tables 2 and 3 show the sentence-level consistency and system-level correlation respectively of TESLA and TESLA-M against the best reported results in WMT 2009 under identical setting. The results show that both TESLA and TESLA-M give very competitive performances. Interestingly, TESLA and TESLA-M obtain similar scores in the out-of-English task. This could be because we use only one pivot language (English), compared to four in the into-English task. We plan to investigate this phenomenon in our future work.

6 Conclusion

This paper describes TESLA-M and TESLA. Our main contributions are: (1) we generalize the bipartite matching formalism of MaxSim into a more expressive linear programming framework;

[6] The original WMT09 report contained erroneous results. The scores here are the corrected results released after publication.

[7] www.wiktionary.org

	en-fr	en-de	en-es	en-cz	Overall
TESLA	0.6828	0.5734	0.5940	0.5519	0.5796
TESLA-M	0.6390	0.5890	0.5927	0.5656	0.5847
wcd6p4er	0.67	0.58	0.61	0.59	0.60
wpF	0.66	0.60	0.61	n/a	0.61
terp	0.62	0.50	0.54	0.31	0.43

Table 2: Out-of-English task sentence-level consistency on WMT 2009 data

	en-fr	en-de	en-es	en-cz	Overall
TESLA	0.8529	0.7857	0.7272	0.3141	0.6700
TESLA-M	0.9294	0.8571	0.7909	0.0857	0.6657
wcd6p4er	-0.89	0.54	-0.45	-0.1	-0.22
wpF	0.90	-0.06	0.58	n/a	n/a
terp	-0.89	0.03	-0.58	-0.40	-0.46

Table 3: Out-of-English task system-level correlation on WMT 2009 data

(2) we exploit parallel texts to create a shallow semantic representation of the sentences; and (3) we show that they outperform all participants in most WMT 2009 shared evaluation tasks.

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

J. Atserias, B. Casas, E. Comelles, M. González, L. Padró, and M. Padró. 2006. Freeling 1.3: Syntactic and semantic services in an open-source NLP library. In *Proceedings of LREC*.

S. Banerjee and A. 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*.

C. Bannard and C. Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of ACL*.

C. Callison-Burch, P. Koehn, and M. Osborne. 2006. Improved statistical machine translation using paraphrases. In *Proceedings of HLT-NAACL*.

C. Callison-Burch, P. Koehn, C. Monz, and J. Schroeder. 2009. Findings of the 2009 Workshop on Statistical Machine Translation. In *Proceedings of WMT*.

Y.S. Chan and H.T. Ng. 2008. MAXSIM: A maximum similarity metric for machine translation evaluation. In *Proceedings of ACL*.

T. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, 2001. *Introduction to Algorithms*. MIT Press, Cambridge, MA.

C. Fellbaum, editor. 1998. *WordNet: An electronic lexical database*. MIT Press, Cambridge, MA.

J. Giménez and L. Màrquez. 2008. A smorgasbord of features for automatic MT evaluation. In *Proceedings of the Third WMT*.

A. Haghighi, J. Blitzer, J. DeNero, and D. Klein. 2009. Better word alignments with supervised ITG models. In *Proceedings of ACL-IJCNLP*.

T. Joachims. 2006. Training linear svms in linear time. In *Proceedings of KDD*.

P. Koehn and K. Knight. 2003. Empirical methods for compound splitting. In *Proceedings of EACL*.

P. Koehn, F.J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT-NAACL*.

P. Liang, B. Taskar, and D. Klein. 2006. Alignment by agreement. In *Proceedings of HLT-NAACL*.

F.J. Och and N. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1).

S. Pado, M. Galley, D. Jurafsky, and C.D. Manning. 2009. Robust machine translation evaluation with entailment features. In *Proceedings of ACL-IJCNLP*.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of ACL*.

M. Popović and H. Ney. 2009. Syntax-oriented evaluation measures for machine translation output. In *Proceedings of WMT*.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of AMTA*.

M. Snover, N. Madnani, B. Dorr, and R. Schwartz. 2009. Fluency, adequacy, or HTER? Exploring different human judgments with a tunable MT metric. In *Proceedings of WMT*.

A. Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings of ICSLP*.