

Character-Level Machine Translation Evaluation for Languages with Ambiguous Word Boundaries

Chang Liu and Hwee Tou Ng
Department of Computer Science
National University of Singapore
13 Computing Drive, Singapore 117417
{liuchan1, nght}@comp.nus.edu.sg

Abstract

In this work, we introduce the TESLA-CELAB metric (Translation Evaluation of Sentences with Linear-programming-based Analysis – Character-level Evaluation for Languages with Ambiguous word Boundaries) for automatic machine translation evaluation. For languages such as Chinese where words usually have meaningful internal structure and word boundaries are often fuzzy, TESLA-CELAB acknowledges the advantage of character-level evaluation over word-level evaluation. By reformulating the problem in the linear programming framework, TESLA-CELAB addresses several drawbacks of the character-level metrics, in particular the modeling of synonyms spanning multiple characters. We show empirically that TESLA-CELAB significantly outperforms character-level BLEU in the English-Chinese translation evaluation tasks.

1 Introduction

Since the introduction of BLEU (Papineni et al., 2002), automatic machine translation (MT) evaluation has received a lot of research interest. The Workshop on Statistical Machine Translation (WMT) hosts regular campaigns comparing different machine translation evaluation metrics (Callison-Burch et al., 2009; Callison-Burch et al., 2010; Callison-Burch et al., 2011). In the WMT shared tasks, many new generation metrics, such as METEOR (Banerjee and Lavie, 2005), TER (Snover et al., 2006), and TESLA (Liu et al., 2010) have consistently outperformed BLEU as judged by the correlations with human judgments.

The research on automatic machine translation evaluation is important for a number of reasons. Automatic translation evaluation gives machine translation researchers a cheap and reproducible way to guide their research and makes it possible to compare machine translation methods across different studies. In addition, machine translation system parameters are tuned by maximizing the automatic scores. Some recent research (Liu et al., 2011) has shown evidence that replacing BLEU by a newer metric, TESLA, can improve the human judged translation quality.

Despite the importance and the research interest on automatic MT evaluation, almost all existing work has focused on European languages, in particular on English. Although many methods aim to be language neutral, languages with very different characteristics such as Chinese do present additional challenges. The most obvious challenge for Chinese is that of word segmentation.

Unlike European languages, written Chinese is not split into words. Segmenting Chinese sentences into words is a natural language processing task in its own right (Zhao and Liu, 2010; Low et al., 2005). However, many different segmentation standards exist for different purposes, such as Microsoft Research Asia (MSRA) for Named Entity Recognition (NER), Chinese Treebank (CTB) for parsing and part-of-speech (POS) tagging, and City University of Hong Kong (CITYU) and Academia Sinica (AS) for general word segmentation and POS tagging. It is not clear which standard is the best in a given scenario.

The only prior work attempting to address the problem of word segmentation in automatic MT evaluation for Chinese that we are aware of is Li et

| | | |
|-----|----------|----------|
| 买 | 伞 | |
| buy | umbrella | |
| | | |
| 买 | 雨 | 伞 |
| buy | umbrella | |
| | | |
| 买 | 雨 | 伞 |
| buy | rain | umbrella |

Figure 1: Three forms of the same expression *buy umbrella* in Chinese

al. (2011). The work compared various MT evaluation metrics (BLEU, NIST, METEOR, GTM, 1 – TER) with different segmentation schemes, and found that treating every single character as a token (character-level MT evaluation) gives the best correlation with human judgments.

2 Motivation

Li et al. (2011) identify two reasons that character-based metrics outperform word-based metrics. For illustrative purposes, we use Figure 1 as a running example in this paper. All three expressions are semantically identical (*buy umbrella*). The first two forms are identical because 雨伞¹ and 伞 are synonyms. The last form is simply an (arguably wrong) alternative segmented form of the second expression.

1. Word-based metrics do not award partial matches, e.g., 买_雨伞 and 买_伞 would be penalized for the mismatch between 雨伞 and 伞. Character-based metrics award the match between characters 伞 and 伞.
2. Character-based metrics do not suffer from errors and differences in word segmentation, so 买_雨伞 and 买_雨_伞 would be judged exactly equal.

Li et al. (2011) conduct empirical experiments to show that character-based metrics consistently outperform their word-based counterparts. Despite that, we observe two important problems for the character-based metrics:

1. Although partial matches are partially awarded, the mechanism breaks down for n-grams where

¹Literally, *rain umbrella*.

$n > 1$. For example, between 买_雨_伞 and 买_伞, higher-order n-grams such as 买_雨 and 雨_伞 still have no match, and will be penalized accordingly, even though 买_雨_伞 and 买_伞 should match exactly. N-grams such as 买_雨 which cross natural word boundaries and are meaningless by themselves can be particularly tricky.

2. Character-level metrics can utilize only a small part of the Chinese synonym dictionary, such as 你 and 您 (*you*). The majority of Chinese synonyms involve more than one character, such as 雨伞 and 伞 (*umbrella*), and 儿童 and 小孩 (*child*).

In this work, we attempt to address both of these issues by introducing TESLA-CELAB, a character-level metric that also models word-level linguistic phenomenon. We formulate the n-gram matching process as a real-valued linear programming problem, which can be solved efficiently. The metric is based on the TESLA automatic MT evaluation framework (Liu et al., 2010; Dahlmeier et al., 2011).

3 The Algorithm

3.1 Basic Matching

We illustrate our matching algorithm using the examples in Figure 1. Let 买雨伞 be the reference, and 买伞 be the candidate translation.

We use Cilin (同义词词林)² as our synonym dictionary. The basic n-gram matching problem is shown in Figure 2. Two n-grams are connected if they are identical, or if they are identified as synonyms by Cilin. Notice that all n-grams are put in the same matching problem regardless of n , unlike in translation evaluation metrics designed for European languages. This enables us to designate n-grams with different values of n as synonyms, such as 雨伞 ($n = 2$) and 伞 ($n = 1$).

In this example, we are able to make a total of two successful matches. The recall is therefore 2/6 and the precision is 2/3.

²http://ir.hit.edu.cn/phpwebsite/index.php?module=pagemaster&PAGE_user_op=view_page&PAGE_id=162

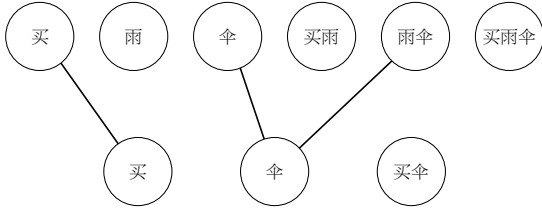


Figure 2: The basic n-gram matching problem

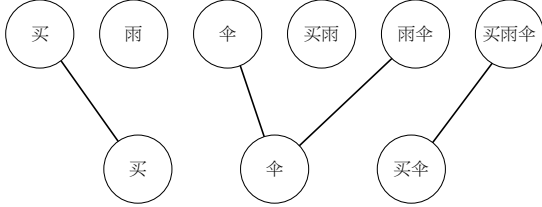


Figure 3: The n-gram matching problem after phrase matching

3.2 Phrase Matching

We note in Figure 2 that the trigram 买雨伞 and the bigram 买伞 are still unmatched, even though the match between 雨伞 and 伞 should imply the match between 买雨伞 and 买伞.

We infer the matching of such phrases using a dynamic programming algorithm. Two n-grams are considered synonyms if they can be segmented into synonyms that are aligned. With this extension, we are able to match 买雨伞 and 买伞 (since 买 matches 买 and 雨伞 matches 伞). The matching problem is now depicted by Figure 3.

The linear programming problem is mathematically described as follows. The variables $w(\cdot, \cdot)$ are the weights assigned to the edges,

$$\begin{aligned} w(\text{买}, \text{买}) &\in [0, 1] \\ w(\text{伞}, \text{伞}) &\in [0, 1] \\ w(\text{雨伞}, \text{伞}) &\in [0, 1] \\ w(\text{买雨伞}, \text{买伞}) &\in [0, 1] \end{aligned}$$

We require that for any node N , the sum of weights assigned to edges linking N must not exceed one.

$$\begin{aligned} w_{\text{ref}}(\text{买}) &= w(\text{买}, \text{买}) \\ w_{\text{ref}}(\text{伞}) &= w(\text{伞}, \text{伞}) \\ w_{\text{ref}}(\text{雨伞}) &= w(\text{雨伞}, \text{伞}) \\ w_{\text{ref}}(\text{买雨伞}) &= w(\text{买雨伞}, \text{买伞}) \end{aligned}$$

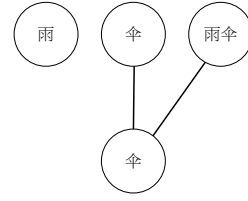


Figure 4: A covered n-gram matching problem

$$\begin{aligned} w_{\text{cand}}(\text{买}) &= w(\text{买}, \text{买}) \\ w_{\text{cand}}(\text{伞}) &= w(\text{伞}, \text{伞}) + w(\text{雨伞}, \text{伞}) \\ w_{\text{cand}}(\text{买伞}) &= w(\text{买雨伞}, \text{买伞}) \end{aligned}$$

where

$$\begin{aligned} w_{\text{ref}}(X) &\in [0, 1] && \forall X \\ w_{\text{cand}}(X) &\in [0, 1] && \forall X \end{aligned}$$

Now we maximize the total match,

$$w(\text{买}, \text{买}) + w(\text{伞}, \text{伞}) + w(\text{雨伞}, \text{伞}) + w(\text{买雨伞}, \text{买伞})$$

In this example, the best match is 3, resulting in a recall of 3/6 and a precision of 3/3.

3.3 Covered Matching

In Figure 3, n-grams 雨 and 买雨 in the reference remain impossible to match, which implies misguided penalty for the candidate translation. We observe that since 买雨伞 has been matched, all its sub-ngrams should be considered matched as well, including 雨 and 买雨. We call this the covered n-gram matching rule. This relationship is implicit in the matching problem for English translation evaluation metrics where words are well delimited. But with phrase matching in Chinese, it must be modeled explicitly.

However, we cannot simply perform covered n-gram matching as a post processing step. As an example, suppose we are matching phrases 雨伞 and 伞, as shown in Figure 4. The linear programming solver may come up with any of the solutions where $w(\text{伞}, \text{伞}) + w(\text{雨伞}, \text{伞}) = 1$, $w(\text{伞}, \text{伞}) \in [0, 1]$, and $w(\text{雨伞}, \text{伞}) \in [0, 1]$.

To give the maximum coverage for the node 雨, only the solution $w(\text{伞}, \text{伞}) = 0$, $w(\text{雨伞}, \text{伞}) = 1$ is accepted. This indicates the need to model covered

n-gram matching in the linear programming problem itself.

We return to the matching of the reference 买雨伞 and the candidate 买伞 in Figure 3. On top of the $w(\cdot)$ variables already introduced, we add the variables *maximum covering weights* $c(\cdot)$. Each $c(X)$ represents the maximum $w(Y)$ variable where n-gram Y completely covers n-gram X .

$$\begin{aligned}
c_{\text{ref}}(\text{买}) &\leq \max(w_{\text{ref}}(\text{买}), w_{\text{ref}}(\text{买雨}), \\
&\quad w_{\text{ref}}(\text{买雨伞})) \\
c_{\text{ref}}(\text{雨}) &\leq \max(w_{\text{ref}}(\text{雨}), w_{\text{ref}}(\text{买雨}), \\
&\quad w_{\text{ref}}(\text{雨伞}), w_{\text{ref}}(\text{买雨伞})) \\
c_{\text{ref}}(\text{伞}) &\leq \max(w_{\text{ref}}(\text{伞}), w_{\text{ref}}(\text{雨伞}), \\
&\quad w_{\text{ref}}(\text{买雨伞})) \\
c_{\text{ref}}(\text{买雨}) &\leq \max(w_{\text{ref}}(\text{买雨}), w_{\text{ref}}(\text{买雨伞})) \\
c_{\text{ref}}(\text{雨伞}) &\leq \max(w_{\text{ref}}(\text{雨伞}), w_{\text{ref}}(\text{买雨伞})) \\
c_{\text{ref}}(\text{买雨伞}) &\leq w_{\text{ref}}(\text{买雨伞}) \\
c_{\text{cand}}(\text{买}) &\leq \max(w_{\text{cand}}(\text{买}), w_{\text{cand}}(\text{买伞})) \\
c_{\text{cand}}(\text{伞}) &\leq \max(w_{\text{cand}}(\text{伞}), w_{\text{cand}}(\text{买伞})) \\
c_{\text{cand}}(\text{买伞}) &\leq w_{\text{cand}}(\text{买伞})
\end{aligned}$$

where

$$\begin{aligned}
c_{\text{ref}}(X) &\in [0, 1] && \forall X \\
c_{\text{cand}}(X) &\in [0, 1] && \forall X
\end{aligned}$$

However, the $\max(\cdot)$ operator is not allowed in the linear programming formulation. We get around this by approximating $\max(\cdot)$ with the sum instead. Hence,

$$\begin{aligned}
c_{\text{ref}}(\text{买}) &\leq w_{\text{ref}}(\text{买}) + w_{\text{ref}}(\text{买雨}) + \\
&\quad w_{\text{ref}}(\text{买雨伞}) \\
c_{\text{ref}}(\text{雨}) &\leq w_{\text{ref}}(\text{雨}) + w_{\text{ref}}(\text{买雨}) + \\
&\quad w_{\text{ref}}(\text{雨伞}) + w_{\text{ref}}(\text{买雨伞}) \\
&\dots
\end{aligned}$$

We justify this approximation by the following observation. Consider the sub-problem consisting of just the $w(\cdot, \cdot)$, $w_{\text{ref}}(\cdot)$, $w_{\text{cand}}(\cdot)$ variables and their associated constraints. This sub-problem can be seen as a maximum flow problem where all constants are integers, hence there exists an optimal solution where each of the w variables is assigned a value of either 0 or 1. For such a solution, the

\max and the sum forms are equivalent, since the $c_{\text{ref}}(\cdot)$ and $c_{\text{cand}}(\cdot)$ variables are also constrained to the range $[0, 1]$.

The maximum flow equivalence breaks down when the $c(\cdot)$ variables are introduced, so in the general case, replacing \max with sum is only an approximation.

Returning to our sample problem, the linear programming solver simply needs to assign:

$$\begin{aligned}
w(\text{买雨伞}, \text{买伞}) &= 1 \\
w_{\text{ref}}(\text{买雨伞}) &= 1 \\
w_{\text{cand}}(\text{买伞}) &= 1
\end{aligned}$$

Consequently, due to the maximum covering weights constraint, we can give the following value assignment, implying that all n-grams have been matched.

$$\begin{aligned}
c_{\text{ref}}(X) &= 1 && \forall X \\
c_{\text{cand}}(X) &= 1 && \forall X
\end{aligned}$$

3.4 The Objective Function

We now define our objective function in terms of the $c(\cdot)$ variables. The recall is a function of $\sum_X c_{\text{ref}}(X)$, and the precision is a function of $\sum_Y c_{\text{cand}}(Y)$, where X is the set of all n-grams of the reference, and Y is the set of all n-grams of the candidate translation.

Many prior translation evaluation metrics such as MAXSIM (Chan and Ng, 2008) and TESLA (Liu et al., 2010; Dahlmeier et al., 2011) use the F-0.8 measure as the final score:

$$F_{0.8} = \frac{\text{Precision} \times \text{Recall}}{0.8 \times \text{Precision} + 0.2 \times \text{Recall}}$$

Under some simplifying assumptions — specifically, that precision = recall — basic calculus shows that $F_{0.8}$ is four times as sensitive to recall than to precision. Following the same reasoning, we want to place more emphasis on recall than on precision. We are also constrained by the linear programming framework, hence we set the objective function as

$$\frac{1}{Z} \left(\sum_X c_{\text{ref}}(X) + f \sum_Y c_{\text{cand}}(Y) \right) \quad 0 < f < 1$$

We set $f = 0.25$ so that our objective function is also four times as sensitive to recall than to precision.³ The value of this objective function is our TESLA-CELAB score. Similar to the other TESLA metrics, when there are N multiple references, we match the candidate translation against each of them and use the average of the N objective function values as the segment level score. System level score is the average of all the segment level scores.

Z is a normalizing constant to scale the metric to the range $[0, 1]$, chosen so that when all the $c(\cdot)$ variables have the value of one, our metric score attains the value of one.

4 Experiments

In this section, we test the effectiveness of TESLA-CELAB on some real-world English-Chinese translation tasks.

4.1 IWSLT 2008 English-Chinese CT

The test set of the IWSLT 2008 (Paul, 2008) English-Chinese ASR challenge task (CT) consists of 300 sentences of spoken language text. The average English source sentence is 5.8 words long and the average Chinese reference translation is 9.2 characters long. The domain is travel expressions.

The test set was translated by seven MT systems, and each translation has been manually judged for adequacy and fluency. Adequacy measures whether the translation conveys the correct meaning, even if the translation is not fully fluent, whereas fluency measures whether a translation is fluent, regardless of whether the meaning is correct. Due to high evaluation costs, adequacy and fluency assessments were limited to the translation outputs of four systems. In addition, the translation outputs of the MT systems are also manually ranked according to their translation quality.

Inter-judge agreement is measured by the Kappa coefficient, defined as:

$$\text{Kappa} = \frac{P(A) - P(E)}{1 - P(E)}$$

where $P(A)$ is the percentage of agreement, and $P(E)$ is the percentage of agreement by pure

³Our empirical experiments suggest that the correlations do plateau near this value. For simplicity, we choose not to tune f on the training data.

| Judgment Set | 2 | 3 |
|--------------|--------|--------|
| 1 | 0.4406 | 0.4355 |
| 2 | - | 0.4134 |

Table 1: Inter-judge Kappa for the NIST 2008 English-Chinese task

chance. The inter-judge Kappa is 0.41 for fluency, 0.40 for adequacy, and 0.57 for ranking. Kappa values between 0.4 and 0.6 are considered *moderate*, and the numbers are in line with other comparable experiments.

4.2 NIST 2008 English-Chinese MT Task

The NIST 2008 English-Chinese MT task consists of 127 documents with 1,830 segments, each with four reference translations and eleven automatic MT system translations. The data is available as LDC2010T01 from the Linguistic Data Consortium (LDC). The domain is newswire texts. The average English source sentence is 21.5 words long and the average Chinese reference translation is 43.2 characters long.

Since no manual evaluation is given for the data set, we recruited twelve bilingual judges to evaluate the first thirty documents for adequacy and fluency (355 segments for a total of $355 \times 11 = 3,905$ translated segments). The final score of a sentence is the average of its adequacy and fluency scores. Each judge works on one quarter of the sentences so that each translation is judged by three judges. The judgments are concatenated to form three full sets of judgments.

We ignore judgments where two sentences are equal in quality, so that there are only two possible outcomes (X is better than Y; or Y is better than X), and $P(E) = 1/2$. The Kappa values are shown in Table 1. The values indicate moderate agreement, and are in line with other comparable experiments.

4.3 Baseline Metrics

4.3.1 BLEU

Although word-level BLEU has often been found inferior to the new-generation metrics when the target language is English or other European languages, prior research has shown that character-level BLEU is highly competitive when the target language is Chinese (Li et al., 2011). Therefore, we

| Metric | Type | Segment consistency | Pearson correlation | Spearman rank correlation |
|--------------|-----------------|---------------------|---------------------|---------------------------|
| BLEU | character-level | 0.7004 | 0.9130 | 0.9643 |
| TESLA-M | word-level | 0.6771 | 0.9167 | 0.8929 |
| TESLA-CELAB– | character-level | 0.7018 | 0.9229 | 0.9643 |
| TESLA-CELAB | hybrid | 0.7281* | 0.9490** | 0.9643 |

Table 2: Correlation with human judgment on the IWSLT 2008 English-Chinese challenge task. * denotes better than the BLEU baseline at 5% significance level. ** denotes better than the BLEU baseline at 1% significance level.

| Metric | Type | Segment consistency | Pearson correlation | Spearman rank correlation |
|--------------|-----------------|---------------------|---------------------|---------------------------|
| BLEU | character-level | 0.7091 | 0.8429 | 0.7818 |
| TESLA-M | word-level | 0.6969 | 0.8301 | 0.8091 |
| TESLA-CELAB– | character-level | 0.7158 | 0.8514 | 0.8227 |
| TESLA-CELAB | hybrid | 0.7162 | 0.8923** | 0.8909** |

Table 3: Correlation with human judgment on the NIST 2008 English-Chinese MT task. ** denotes better than the BLEU baseline at 1% significance level.

use character-level BLEU as our main baseline.

The correlations of character-level BLEU and the average human judgments are shown in the first row of Tables 2 and 3 for the IWSLT and the NIST data set, respectively. Segment-level consistency is defined as the number of correctly predicted pairwise rankings divided by the total number of pairwise rankings. Ties are excluded from the calculation. We also report the Pearson correlation and the Spearman rank correlation of the system-level scores. Note that in the IWSLT data set, the Spearman rank correlation is highly unstable due to the small number of participating systems.

4.3.2 TESLA-M

In addition to character-level BLEU, we also present the correlations for the word-level metric TESLA. Compared to BLEU, TESLA allows more sophisticated weighting of n-grams and measures of word similarity including synonym relations. It has been shown to give better correlations than BLEU for many European languages including English (Callison-Burch et al., 2011). However, its use of POS tags and synonym dictionaries prevents its use at the character-level. We use TESLA as a representative of a competitive word-level metric.

We use the Stanford Chinese word segmenter (Tseng et al., 2005) and POS tagger (Toutanova et al., 2003) for preprocessing and Cilin for synonym

definition during matching. TESLA has several variants, and the simplest and often the most robust, TESLA-M, is used in this work. The various correlations are reported in the second row of Tables 2 and 3.

The scores show that word-level TESLA-M has no clear advantage over character-level BLEU, despite its use of linguistic features. We consider this conclusion to be in line with that of Li et al. (2011).

4.4 TESLA-CELAB

In all our experiments here we use TESLA-CELAB with n-grams for n up to four, since the vast majority of Chinese words, and therefore synonyms, are at most four characters long.

The correlations between the TESLA-CELAB scores and human judgments are shown in the last row of Tables 2 and 3. We conducted significance testing using the resampling method of (Koehn, 2004). Entries that outperform the BLEU baseline at 5% significance level are marked with ‘*’, and those that outperform at the 1% significance level are marked with ‘**’. The results indicate that TESLA-CELAB significantly outperforms BLEU.

For comparison, we also run TESLA-CELAB without the use of the Cilin dictionary, reported in the third row of Tables 2 and 3 and denoted as TESLA-CELAB–. This disables TESLA-

CELAB’s ability to detect word-level synonyms and turns TESLA-CELAB into a linear programming based character-level metric. The performance of TESLA-CELAB— is comparable to the character-level BLEU baseline.

Note that

- TESLA-M can process word-level synonyms, but does not award character-level matches.
- TESLA-CELAB— and character-level BLEU award character-level matches, but do not consider word-level synonyms.
- TESLA-CELAB can process word-level synonyms and can award character-level matches.

Therefore, the difference between TESLA-M and TESLA-CELAB highlights the contribution of character-level matching, and the difference between TESLA-CELAB— and TESLA-CELAB highlights the contribution of word-level synonyms.

4.5 Sample Sentences

Some sample sentences taken from the IWSLT test set are shown in Table 4 (some are simplified from the original). The Cilin dictionary correctly identified the following as synonyms:

| | | | |
|----|---|----|-----------------|
| 周 | = | 星期 | <i>week</i> |
| 女儿 | = | 闺女 | <i>daughter</i> |
| 你 | = | 您 | <i>you</i> |
| 工作 | = | 上班 | <i>work</i> |

The dictionary fails to recognize the following synonyms:

| | | | |
|----|---|----|-------------|
| 一个 | = | 个 | <i>a</i> |
| 这儿 | = | 这里 | <i>here</i> |

However, partial awards are still given for the matching characters 这 and 个.

Based on these synonyms, TESLA-CELAB is able to award less trivial n-gram matches, such as 下周=下星期, 个女儿=个闺女, and 工作吗=上班吗, as these pairs can all be segmented into aligned synonyms. The covered n-gram matching rule is then able to award tricky n-grams such as 下星, 个女, 个闺, 作吗 and 班吗, which are covered by 下星期, 个女儿, 个闺女, 工作吗 and 上班吗 respectively.

Note also that the word segmentations shown in these examples are for clarity only. The TESLA-CELAB algorithm does not need pre-segmented

| | | | |
|------------|------|------|---|
| Reference: | 下 | 周 | 。 |
| | next | week | . |
| Candidate: | 下 | 星期 | 。 |
| | next | week | . |

| | | | | | |
|------------|---|------|----|----------|---|
| Reference: | 我 | 有 | 一个 | 女儿 | 。 |
| | I | have | a | daughter | . |
| Candidate: | 我 | 有 | 个 | 闺女 | 。 |
| | I | have | a | daughter | . |

| | | | | | | |
|------------|-----|----|------|------|----|---|
| Reference: | 你 | 在 | 这儿 | 工作 | 吗 | ？ |
| | you | at | here | work | qn | ? |
| Candidate: | 您 | 在 | 这里 | 上班 | 吗 | ？ |
| | you | at | here | work | qn | ? |

Table 4: Sample sentences from the IWSLT 2008 test set

| | |
|-----------------|------------|
| Schirm | kaufen |
| <i>umbrella</i> | <i>buy</i> |

| | |
|-----------------|------------|
| Regenschirm | kaufen |
| <i>umbrella</i> | <i>buy</i> |

| | | |
|-------------|-----------------|------------|
| Regen | schirm | kaufen |
| <i>rain</i> | <i>umbrella</i> | <i>buy</i> |

Figure 5: Three forms of *buy umbrella* in German

sentences, and essentially finds multi-character synonyms opportunistically.

5 Discussion and Future Work

5.1 Other Languages with Ambiguous Word Boundaries

Although our experiments here are limited to Chinese, many other languages have similarly ambiguous word boundaries. For example, in German, the exact counterpart to our example exists, as depicted in Figure 5.

Regenschirm, literally *rain-umbrella*, is a synonym of Schirm. The first two forms in Figure 5 appear in natural text, and in standard BLEU, they would be penalized for the non-matching words *Schirm* and *Regenschirm*. Since compound nouns such as *Regenschirm* are very common in German and generate many out-of-vocabulary words, a common preprocessing step in German translation (and translation evaluation to a lesser extent) is to split compound words, and we end up with the last form *Regen schirm kaufen*. This process is analogous to

Chinese word segmentation.

We plan to conduct experiments on German and other Asian languages with the same linguistic phenomenon in future work.

5.2 Fractional Similarity Measures

In the current formulation of TESLA-CELAB, two n-grams X and Y are either synonyms which completely match each other, or are completely unrelated. In contrast, the linear-programming based TESLA metric allows fractional similarity measures between 0 (completely unrelated) and 1 (exact synonyms). We can then award partial scores for related words, such as those identified as such by WordNet or those with the same POS tags.

Supporting fractional similarity measures is non-trivial in the TESLA-CELAB framework. We plan to address this in future work.

5.3 Fractional Weights for N-grams

The TESLA-M metric allows each n-gram to have a weight, which is primarily used to discount function words. TESLA-CELAB can support fractional weights for n-grams as well by the following extension. We introduce a function $m(X)$ that assigns a weight in $[0, 1]$ for each n-gram X . Accordingly, our objective function is replaced by:

$$\frac{1}{Z} \left(\sum_X m(X)c_{\text{ref}}(X) + f \sum_Y m(Y)c_{\text{cand}}(Y) \right)$$

where Z is a normalizing constant so that the metric has a range of $[0, 1]$.

$$Z = \sum_X m(X) + f \sum_Y m(Y)$$

However, experiments with different weight functions $m(\cdot)$ on the test data set failed to find a better weight function than the currently implied $m(\cdot) = 1$. This is probably due to the linguistic characteristics of Chinese, where human judges apparently give equal importance to function words and content words. In contrast, TESLA-M found discounting function words very effective for English and other European languages such as German. We plan to investigate this in the context of non-Chinese languages.

6 Conclusion

In this work, we devise a new MT evaluation metric in the family of TESLA (Translation Evaluation of Sentences with Linear-programming-based Analysis), called TESLA-CELAB (Character-level Evaluation for Languages with Ambiguous word Boundaries), to address the problem of fuzzy word boundaries in the Chinese language, although neither the phenomenon nor the method is unique to Chinese. Our metric combines the advantages of character-level and word-level metrics:

1. TESLA-CELAB is able to award scores for partial word-level matches.
2. TESLA-CELAB does not have a segmentation step, hence it will not introduce word segmentation errors.
3. TESLA-CELAB is able to take full advantage of the synonym dictionary, even when the synonyms differ in the number of characters.

We show empirically that TESLA-CELAB significantly outperforms the strong baseline of character-level BLEU in two well known English-Chinese MT evaluation data sets. The source code of TESLA-CELAB is available from <http://nlp.comp.nus.edu.sg/software/>.

Acknowledgments

This research is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

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*.
- Chris Callison-Burch, Philipp Koehn, Christof Monz, and Josh Schroeder. 2009. Findings of the 2009 workshop on statistical machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*.

- Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, Mark Przybocki, and Omar F. Zaidan. 2010. Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*.
- Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar Zaidan. 2011. Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*.
- 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: Human Language Technologies*.
- Daniel Dahlmeier, Chang Liu, and Hwee Tou Ng. 2011. TESLA at WMT2011: Translation evaluation and tunable metric. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*.
- Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*.
- Maoxi Li, Chengqing Zong, and Hwee Tou Ng. 2011. Automatic evaluation of Chinese translation output: word-level or character-level? In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Short Papers*.
- 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*.
- Jin Kiat Low, Hwee Tou Ng, and Wenyuan Guo. 2005. A maximum entropy approach to Chinese word segmentation. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*.
- 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*.
- Michael Paul. 2008. Overview of the iwslt 2008 evaluation campaign. In *Proceedings of the International Workshop on Spoken Language Translation*.
- Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the Association for Machine Translation in the Americas*.
- Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics*.
- Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A conditional random field word segmenter for SIGHAN bakeoff 2005. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*.
- Hongmei Zhao and Qun Liu. 2010. The CIPS-SIGHAN CLP 2010 Chinese word segmentation bakeoff. In *Proceedings of the Joint Conference on Chinese Language Processing*.