

MAXSIM: An Automatic Metric for Machine Translation Evaluation Based on Maximum Similarity

Yee Seng Chan and Hwee Tou Ng

Department of Computer Science

National University of Singapore

Law Link, Singapore 117590

{chanys, nght}@comp.nus.edu.sg

Abstract

This paper describes our participation in the NIST 2008 MetricsMATR Challenge, using our recently proposed automatic machine translation evaluation metric MAXSIM. The metric calculates a similarity score between a pair of English system-reference sentences by comparing information items such as n-grams across the sentence pair. Unlike most metrics, MAXSIM computes a similarity score between items, then find a maximum weight matching between the items such that each item in one sentence is mapped to at most one item in the other sentence. Evaluation on the WMT07, WMT08, and MT06 datasets show that MAXSIM achieves good correlations with human judgment.

1 Introduction

An important contributing factor to the success of statistical machine translation (MT) research in recent years is the introduction of BLEU (Papineni et al., 2002), which is an automatic MT evaluation metric. Having an automatic metric is important as it alleviates the need for human judgment of MT output. This allows rapid testing of revisions made during the development of an MT system.

Although BLEU is widely popular and has contributed significantly to the progress of MT research, it is becoming evident that BLEU does not correlate with human judgment well enough and suffers from several other deficiencies (Chiang et al., 2008). Having recognized the limitations of BLEU, conducting research to propose novel automatic MT evalua-

tion metrics is becoming increasingly popular. During the recent second workshop on statistical MT (WMT07) (Callison-Burch et al., 2007), 11 automatic MT evaluation metrics were evaluated for correlation with human judgement for translation into English. Following that, the third workshop on statistical MT (WMT08) (Callison-Burch et al., 2008) evaluated 14 automatic MT evaluation metrics. Results of the workshops show that BLEU lags behind several other metrics in terms of correlation with human judgment.

This paper describes our participation in the NIST 2008 MetricsMATR Challenge, using our recently proposed automatic MT evaluation metric MAXSIM (Chan and Ng, 2008). To compute a similarity score between a pair of English system-reference sentences, MAXSIM extracts and compares information items such as n-grams. Since different concepts can be expressed in a language in many ways, MAXSIM allows matching via a word's lemmatized form and synonyms. The grammaticality and general fluency of a translation are also checked by incorporating sequences of part-of-speech (POS) tags during the matching process. Finally, in contrast to current metrics which perform binary matching (either a pair of information items match, or they do not), MAXSIM computes a similarity score between a pair of items. Doing this means that there are many possible ways to match information items across a pair of system-reference sentences, with each match having its own similarity weight. To match each system item to at most one reference item, we model the items in the sentence pair as nodes in a bipartite graph and use the Kuhn-Munkres algorithm (Kuhn,

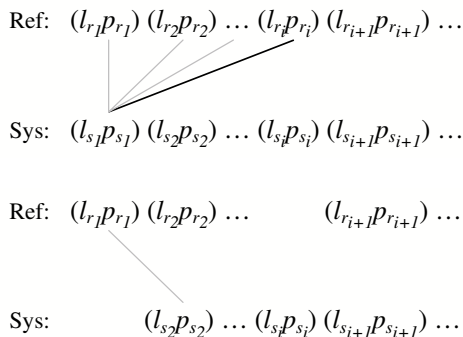


Figure 1: Matching n-grams.

1955; Munkres, 1957) to find a *maximum* weight matching between the items in polynomial time. The weights of the edges of the resulting graph will then contribute towards the final similarity score of the sentence pair. In (Chan and Ng, 2008), we show that MAXSIM achieves higher correlation with human judgment than all of the 11 automatic MT evaluation metrics evaluated during WMT07 (Callison-Burch et al., 2007).

2 The Maximum Similarity Metric

We now describe our proposed metric MAXSIM, which is based on precision and recall, allows for synonyms, and weights the matches found.

To compute similarity scores, MAXSIM requires a number of resources. Given a pair of English sentences to be compared (a system translation against a reference translation), we perform tokenization¹, lemmatization using WordNet-2.1², and part-of-speech (POS) tagging with the MXPOST tagger (Ratnaparkhi, 1996). Next, we remove all non-alphanumeric tokens. We also gather the set of WordNet-2.1 synonyms for each word (noun, verb, adjective, and adverb). These will be used when computing similarity scores between item pairs.

2.1 Matching Using N-gram Information

To calculate a similarity score for a pair of system-reference translation sentences, MAXSIM extracts and compares n-gram information. In our work, we use unigrams, bigrams, and trigrams. Based on these comparisons or matches of n-grams across the

sentence pair, MAXSIM computes the corresponding scores for precision and recall. These are then combined into parameterized F scores.

To match n-grams, MAXSIM goes through a sequence of three phases: lemma and POS matching, lemma matching, and bipartite graph matching. In this section, we will first illustrate the matching process using unigrams, then describe the extension to bigrams and trigrams.

Lemma and POS matching Representing each n-gram by its sequence of lemma and POS tag pairs, we first try to perform an exact match in both lemma and POS tag. In all our n-gram matching, each n-gram in the system translation can only match at most *one* n-gram in the reference translation.

Representing each unigram ($l_i p_i$) at position i by its lemma l_i and POS tag p_i , we accumulate the number of matching system-reference unigram pairs in $match_{uni}$. To find matching pairs, we proceed in a left-to-right fashion (in both strings). As the top part of Figure 1 illustrates, we take the first system unigram and check if it matches the first reference unigram. If they do not match, we then check against the second reference unigram and so on until we find a match. A system unigram ($l_{s_i} p_{s_i}$) matches a reference unigram ($l_{r_j} p_{r_j}$) if $l_{s_i} = l_{r_j}$ and $p_{s_i} = p_{r_j}$. The example given in the top half of Figure 1 shows that we have found a match between the first system unigram and the i th reference unigram. Once there is a match, we increment $match_{uni}$ by 1 and remove the pair of system-reference unigrams from further consideration (removed items will not be matched again subsequently). Hence, each system item can match at most one reference item. Then, as shown in the bottom half of Figure 1, we next try to match the second system unigram against the reference unigrams, once again proceeding in a left-to-right fashion. If no match is found for a particular system unigram, we move on to the next system unigram. We continue this process until we complete the matching of the last system unigram.

Lemma matching For the remaining set of unigrams that are not yet matched, we now relax our matching criteria by allowing a match if their corresponding lemmas match. That is, a system unigram ($l_{s_i} p_{s_i}$) matches a reference unigram ($l_{r_j} p_{r_j}$) if $l_{s_i} = l_{r_j}$. Once again, we find matches in a left-to-right fashion and add the number of unigram

¹<http://www.cis.upenn.edu/~treebank/tokenizer.sed>

²<http://wordnet.princeton.edu/man/morph.3WN>

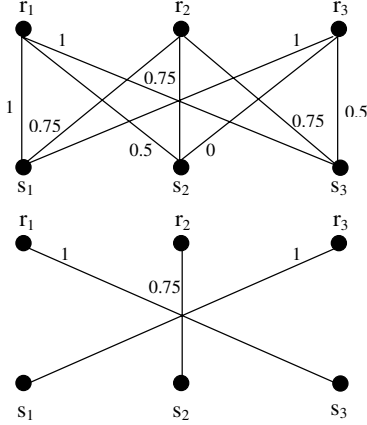


Figure 2: Bipartite matching.

matches found during this phase to $match_{uni}$.

Bipartite graph matching For the remaining unigrams that are not matched so far, we try to match them by constructing a weighted complete bipartite graph. Each of these remaining unigrams will be represented as a node in the graph. Note that, without loss of generality, if the number of system nodes and reference nodes are not the same, we can simply add dummy nodes with connecting edges of weight 0 to obtain a complete bipartite graph with equal number of nodes on both sides. Also, the graph is weighted in the sense that each graph edge e connecting a pair of system-reference unigrams has a weight $w(e)$, which indicates the degree of similarity between the unigram pair. As an example, we show in the top half of Figure 2 a complete bipartite graph, constructed for a set of three system unigrams (s_1, s_2, s_3) and three reference unigrams (r_1, r_2, r_3), and the weight of the connecting edge between two unigrams represents their degree of similarity.

We now describe how to calculate the weight $w(e)$ of an edge e . Assume that we have an edge connecting a system unigram ($l_{s_i} p_{s_i}$) to a reference unigram ($l_{r_j} p_{r_j}$). We will calculate a score S of this pair as follows and this will be the weight $w(e)$ of the connecting edge:

$$\begin{aligned} w(e) &= S \\ S &= \frac{I(p_{s_i}, p_{r_j}) + Syn(l_{s_i}, l_{r_j})}{2} \end{aligned} \quad (1)$$

where functions $I(\cdot)$ and $Syn(\cdot)$ are defined as fol-

lows:

$$\begin{aligned} I(p_{s_i}, p_{r_j}) &= \begin{cases} 1, & \text{if } p_{s_i} = p_{r_j} \\ 0, & \text{otherwise} \end{cases} \\ Syn(l_{s_i}, l_{r_j}) &= \begin{cases} 1, & WN_{syn}(l_{s_i}) \cap WN_{syn}(l_{r_j}) \\ & \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

$I(\cdot)$ is an identity function and in this case evaluates to 1 if the two POS tags p_{s_i} and p_{r_j} are identical. The function $Syn(\cdot)$ checks whether l_{s_i} is a synonym of l_{r_j} . To determine this, we obtain the set $WN_{syn}(l_{s_i})$ of WordNet synonyms for l_{s_i} and the set $WN_{syn}(l_{r_j})$ of WordNet synonyms for l_{r_j} . In gathering the set WN_{syn} for a word, we gather all the synonyms for all its senses and do not restrict to a particular POS category.

Once we have calculated the weights of all edges, we will have a weighted complete bipartite graph. However, we want to find the set of edges such that each system unigram is matched or aligned to exactly one reference unigram, while giving a maximum weight matching. This *maximum weighted bipartite matching* problem can be solved in $O(n^3)$ time (where n refers to the number of nodes, or vertices in the graph) using the Kuhn-Munkres algorithm (Kuhn, 1955; Munkres, 1957). The bottom half of Figure 2 shows the resulting maximum weighted bipartite graph, where the alignment represents the maximum weight matching, out of all possible alignments.

Once we have solved and obtained a maximum weight matching M for the unigram bipartite graph, we sum up the weights of the edges to obtain the weight of the matching M :

$$w(M) = \sum_{e \in M} w(e)$$

We then add $w(M)$ to $match_{uni}$.

2.1.1 Calculating F Scores Based on N-gram Matches

Finally, after going through the three phases, we use $match_{uni}$ to calculate the corresponding unigram precision P and unigram recall R . These are

then combined into a parameterized F score:

$$\begin{aligned} P &= \frac{\text{match}_{uni}}{\text{no. of unigrams in system translation}} \\ R &= \frac{\text{match}_{uni}}{\text{no. of unigrams in reference translation}} \\ F &= \frac{P \cdot R}{\alpha P + (1 - \alpha)R} \end{aligned} \quad (2)$$

2.1.2 Extension to Bigrams and Trigrams

We have described the matching and scoring process for unigrams. Similarly, we also count the number match_{bi} of bigram matches and match_{tri} of trigram matches. Based on match_{bi} and match_{tri} , we then calculate the corresponding F scores for bigrams and trigrams.

Since n-grams such as bigrams and trigrams consist of more than one token, we calculate the weight of an edge in a slightly different manner, as compared to the case of unigrams. To calculate the weight $w(e)$ of an edge e connecting a system n-gram $(l_{s_i^1}p_{s_i^1}, \dots, l_{s_i^n}p_{s_i^n})$ and a reference n-gram $(l_{r_j^1}p_{r_j^1}, \dots, l_{r_j^n}p_{r_j^n})$, we calculate a score S_k between each corresponding token of the n-gram pair, using Equation 1. For instance, S_1 corresponds to the score between $l_{s_i^1}p_{s_i^1}$ and $l_{r_j^1}p_{r_j^1}$. These scores are then averaged to obtain the weight of the edge:

$$w(e) = \frac{1}{n} \sum_{k=1}^n S_k$$

Further, we impose an additional condition: $S_k \neq 0$, for $1 \leq k \leq n$, else we will set $w(e) = 0$. This captures the intuition that in matching a system n-gram against a reference n-gram, where $n > 1$, we require each system token to have at least some degree of similarity with the corresponding reference token.

2.2 Scoring a Sentence Pair and the Whole Corpus

To calculate the similarity score of a system-reference sentence pair s , we extract and match the unigrams, bigrams, and trigrams of the sentences. Based on the matches, we then calculate their corresponding F scores. These are then averaged to obtain a single similarity score $score_s$ for the sentence

pair s :

$$score_s = \frac{1}{N} \sum_{n=1}^N F_{s,n}$$

where $F_{s,n}$ denotes the F score for n-grams. For example, $F_{s,2}$ is computed based on the number match_{bi} of bigram matches. In our experiments, we set $N = 3$, using unigram, bigram, and trigram scores. Then, to obtain a single similarity score $sim\text{-score}$ for the entire system corpus, we repeat this process of calculating a $score_s$ for each system-reference sentence pair s , and compute the arithmetic mean over all $|S|$ sentence pairs:

$$sim\text{-score} = \frac{1}{|S|} \sum_{s=1}^{|S|} score_s$$

If we are given access to multiple references, we calculate an individual $sim\text{-score}$ between the system corpus and *each* reference corpus, and then compute their arithmetic mean.

The MAXSIM metric described here, which was submitted to NIST 2008 MetricsMATR Challenge, refers to the MAXSIM_n version as described in (Chan and Ng, 2008). In that paper, we also propose another version MAXSIM_{n+d} which includes the use of dependency relations. We omit the use of dependency relations for this evaluation, since our experiments indicate that adding dependency relations does not provide a substantial gain in correlation scores, while incurring a significant computational cost of determining the dependency relations.

3 Evaluation Data

To evaluate our metric, we conduct experiments on datasets from WMT07, WMT08, and NIST MT06.

3.1 WMT07 Workshop

The ACL-07 workshop on statistical machine translation (WMT07) evaluated the translation performance of MT systems and included a task of measuring the correlation with human judgement of 11 automatic MT evaluation metrics. WMT07 used a Europarl dataset (2,000 English sentences) and a News Commentary dataset (2,007 English sentences), together with their corresponding translations in various languages. As part of the workshop, correlations of the automatic metrics were measured

Dataset	No. of texts	No. of segments	
		Per text	Total
WMT07 Europarl	23	2,000	86,140
WMT07 News	20	2,007	
WMT08 Europarl	38	2,000	151,887
WMT08 News	37	2,051	
MT06	9	249	2,241

Table 1: Size of each dataset, in number of segments (sentences).

for the tasks of translating French, German, and Spanish into English. Hence, we will measure the correlation of MAXSIM on these tasks. Note that for each corpus, there is only one English reference text.

For human evaluation of the MT submissions, WMT07 used a total of four different criteria: adequacy, fluency, rank (we will refer to this as *preference* in this paper), and constituent. In this paper, we will focus on **adequacy** (how much of the original meaning is expressed in a system translation) and **preference** (different translations of a single source sentence are compared and ranked from best to worst), since these are the criteria used in the NIST 2008 MetricsMATR Challenge.

We show in Table 1 the size (in terms of number of sentences) of the WMT07 dataset. Note that WMT07 has only one reference text. For example, the WMT07 Europarl dataset consists of 22 system translation texts and 1 reference text (each having 2,000 sentences), while the WMT07 News Commentary dataset consists of 19 system translation texts and 1 reference text (each having 2,007 sentences). Hence, the WMT07 dataset consists of 86,140 sentences altogether.

3.2 WMT08 Workshop

Based on the success of WMT07, the ACL-08 workshop on statistical machine translation (WMT08) was recently organized and it similarly included a task of measuring the correlations of automatic MT evaluation metrics. WMT08 attracted participation from 14 automatic MT evaluation metrics. WMT08 used a Europarl dataset (2,000 English sentences) and a News dataset (2,051 English sentences) and also measured the correlations of the automatic metrics for the tasks of translating French, German, and

Spanish into English. Hence, we will measure the correlation of MAXSIM on these tasks.

The organizers of WMT08 noted that in WMT07, Kappa values measured for inter- and intra-annotator agreement for adequacy and fluency were substantially lower than those for preference and constituent. This indicated that preference and constituent are more reliable criteria for MT evaluation. Hence, only these two evaluation criteria were used in WMT08. As such, in this paper, we will only use the preference criterion. We show in Table 1 the size of the WMT08 data. Similar to WMT07, WMT08 has only one reference text.

3.3 NIST MT06

As development data for the NIST 2008 MetricsMATR Challenge, some data from the NIST Open MT 2006 evaluation (MT06) and the DARPA TransTac program (Transtac) were released to registered participants. For human evaluation of the MT system translations, two criteria are used: adequacy and preference.

The MT06 data consists of 8 system translations and 4 English references, where each text consists of 249 segments. A segment of text usually consists of a single sentence. Although several reference texts are provided, human judgment for adequacy and preference are only based on the third English reference text. Hence when applying MAXSIM on the MT06 data, we will only use the third English reference text. We show in Table 1 the size of the MT06 data.

For the Transtac data, it consists of 5 system translations and 4 English references, where each text consists of 16 segments. Given the very small size of this data as compared to WMT07, WMT08, and MT06, we do not report results on Transtac in this paper.

4 Results

In this section, we describe the correlation results of MAXSIM. We follow the WMT07 and WMT08 process of converting the system-level raw scores assigned by an automatic metric to ranks and then using the Spearman’s rank correlation coefficient to measure correlation.

In our initial work on MAXSIM (Chan and Ng,

Dataset	MAXSIM			S1			BLEU		
	Adq	Pref	Avg	Adq	Pref	Avg	Adq	Pref	Avg
WMT07	0.804	0.893	0.849	0.774	0.804	0.789	0.690	0.672	0.681
WMT08	–	0.801	0.801	–	0.833	0.833	–	0.520	0.520
MT06	0.976	0.952	0.964	–	–	–	0.643	0.619	0.631

Table 2: System-level correlations of MAXSIM, state-of-the-art automatic MT evaluation metrics (S1), and BLEU on the various datasets.

2008), we simply use $\alpha = 0.9$ in Equation 2 without tuning on any dataset (having an α of more than 0.5 weights recall more than precision). Following this setup, we show in Table 2 the system-level correlations of MAXSIM (using $\alpha = 0.9$) on the various datasets for the adequacy (Adq) and preference (Pref) criteria, as well as their average (Avg). Note that the workshop papers of WMT07 (Callison-Burch et al., 2007) and WMT08 (Callison-Burch et al., 2008) provide the correlation results of the participating metrics in the various tasks. Hence, for comparison, we gather from the workshop papers the correlation results of the top performing metric and show them under the column *S1* in Table 2. For WMT07, the metric based on semantic role overlap (Giménez and Màrquez, 2007) achieves the best correlation for both adequacy and preference. For WMT08, the metric based on dependency relations (Giménez and Màrquez, 2008) achieves the best correlation for preference. For the MT06 data, it has not been used for any MT metric evaluation. To get an indication of how a state-of-the-art automatic MT evaluation metric would perform, we run METEOR (Agarwal and Lavie, 2008) on the MT06 data, obtaining correlation results of 0.905 for adequacy and 0.881 for preference (for an average correlation of 0.893). Also, under the column *BLEU* of Table 2, we show the correlation results of the BLEU metric (version 11b).

We tried tuning the value of α from 0 to 1, in increment of 0.05, and measuring the corresponding correlation scores on the WMT07, WMT08, and MT06 datasets. From our experiments, we find that using $\alpha = 0.8$ gives slightly better overall performance on the three datasets. The average (over adequacy and preference) correlation scores are 0.828 on WMT07, and 0.964 on MT06. On WMT08, the correlation score for preference is 0.835. Hence, we submitted our MAXSIM metric to the NIST 2008

MetricsMATR Challenge using $\alpha = 0.8$. Overall, as our experiments show, MAXSIM is able to achieve competitive correlation scores on benchmark datasets when compared to other state-of-the-art automatic MT evaluation metrics.

Besides achieving good correlation results, another important factor in designing a metric is the time needed to apply the metric. The relatively fast computation of the BLEU metric is an important factor in its popularity. Hence, we also record the time taken to apply MAXSIM on the datasets. To apply MAXSIM, we need to first preprocess the input files (performing tokenization, POS tagging, and lemmatization), before computing the MAXSIM metric score on the processed files. Our experiments indicate that when given a pair of system-reference files of 2,000 sentences each (about 59K words of reference text after tokenization) from the WMT07 Europarl dataset, it takes 48 seconds to preprocess the two files and 19 seconds to compute the MAXSIM metric score, giving a throughput of scoring 30 sentences per second. These timings are obtained on a Linux machine with dual-core 2.2GHz CPU and 2GB of RAM with no parallelization.

5 Conclusion

This paper describes our participation in the NIST 2008 MetricsMATR Challenge, using our recently proposed automatic MT evaluation metric MAXSIM. Evaluation on the datasets of WMT07, WMT08, and MT06 shows that MAXSIM achieves state-of-the-art correlation results.

Acknowledgments

This research is partially supported by research grant POD0713875.

References

- Abhaya Agarwal and Alon Lavie. 2008. Meteor, M-BLEU and M-TER: Evaluation metrics for high-correlation with human rankings of machine translation output. In *Proceedings of the Third Workshop on Statistical Machine Translation, ACL-08:HLT*, pages 115–118.
- Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2007. (Meta-) evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation, ACL-07*, pages 136–158.
- Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2008. Further meta-evaluation of machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation, ACL-08:HLT*, pages 70–106.
- Yee Seng Chan and Hwee Tou Ng. 2008. MAXSIM: A maximum similarity metric for machine translation evaluation. In *Proceedings of ACL-08:HLT*, pages 55–62.
- David Chiang, Steve Deneefe, Yee Seng Chan, and Hwee Tou Ng. 2008. Decomposability of translation metrics for improved evaluation and efficient algorithms. In *Proceedings of EMNLP-08*.
- Jesús Giménez and Lluís Màrquez. 2007. Linguistic features for automatic evaluation of heterogeneous MT systems. In *Proceedings of the Second Workshop on Statistical Machine Translation, ACL-07*, pages 256–264.
- Jesús Giménez and Lluís Màrquez. 2008. A smorgasbord of features for automatic MT evaluation. In *Proceedings of the Third Workshop on Statistical Machine Translation, ACL-08:HLT*, pages 195–198.
- Harold W. Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2(1):83–97.
- James Munkres. 1957. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of ACL-02*, pages 311–318.
- Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of EMNLP-96*, pages 133–142.