MaxSim: A Maximum Similarity Metric for Machine Translation Evaluation

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

We propose an automatic machine translation (MT) evaluation metric that calculates a similarity score (based on precision and recall) of a pair of sentences. Unlike most metrics, we compute a similarity score between items across the two sentences. We 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. This general framework allows us to use arbitrary similarity functions between items, and to incorporate different information in our comparison, such as n-grams, dependency relations, etc. When evaluated on data from the ACL-07 MT workshop, our proposed metric achieves higher correlation with human judgements than all 11 automatic MT evaluation metrics that were evaluated during the workshop.

1 Introduction

In recent years, machine translation (MT) research has made much progress, which includes the introduction of automatic metrics for MT evaluation. Since human evaluation of MT output is time consuming and expensive, having a robust and accurate automatic MT evaluation metric that correlates well with human judgement is invaluable.

Among all the automatic MT evaluation metrics, BLEU (Papineni et al., 2002) is the most widely used. Although BLEU has played a crucial role in the progress of MT research, it is becoming evident that BLEU does not correlate with human judgement well enough, and suffers from several other deficiencies such as the lack of an intuitive interpretation of its scores.

During the recent ACL-07 workshop on statistical MT (Callison-Burch et al., 2007), a total of 11 automatic MT evaluation metrics were evaluated for correlation with human judgement. The results show that, as compared to BLEU, several recently proposed metrics such as Semantic-role overlap (Gimenez and Marquez, 2007), ParaEval-recall (Zhou et al., 2006), and METEOR (Banerjee and Lavie, 2005) achieve higher correlation.

In this paper, we propose a new automatic MT evaluation metric, MaxSim, that compares a pair of system-reference sentences by extracting n-grams and dependency relations. Recognizing that different concepts can be expressed in a variety of ways, we allow matching across synonyms and also compute a score between two matching items (such as between two n-grams or between two dependency relations), which indicates their degree of similarity with each other.

Having weighted matches between items means that there could be many possible ways to match, or link items from a system translation sentence to a reference translation sentence. 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, 1955; Munkres, 1957) to find a *maximum* weight matching (or alignment) between the items in polynomial time. The weights (from the edges) of the resulting graph will then be added to determine the final similarity score between the pair of sentences.

Proceedings of ACL-08: HLT, pages 55–62,
Columbus, Ohio, USA, June 2008. ©2008 Association for Computational Linguistics

Although a maximum weight bipartite graph was also used in the recent work of (Taskar et al., 2005), their focus was on learning supervised models for single word alignment between sentences from a source and target language.

The contributions of this paper are as follows. Current metrics (such as BLEU, METEOR, Semantic-role overlap, ParaEval-recall, etc.) do not assign different weights to their matches: either two items match, or they don't. Also, metrics such as METEOR determine an alignment between the items of a sentence pair by using heuristics such as the least number of matching crosses. In contrast, we propose weighting different matches differently, and then obtain an optimal set of matches, or alignments, by using a maximum weight matching framework. We note that this framework is not used by any of the 11 automatic MT metrics in the ACL-07 MT workshop. Also, this framework allows for defining arbitrary similarity functions between two matching items, and we could match arbitrary concepts (such as dependency relations) gathered from a sentence pair. In contrast, most other metrics (notably BLEU) limit themselves to matching based only on the surface form of words. Finally, when evaluated on the datasets of the recent ACL-07 MT workshop (Callison-Burch et al., 2007), our proposed metric achieves higher correlation with human judgements than all of the 11 automatic MT evaluation metrics evaluated during the workshop.

In the next section, we describe several existing metrics. In Section 3, we discuss issues to consider when designing a metric. In Section 4, we describe our proposed metric. In Section 5, we present our experimental results. Finally, we outline future work in Section 6, before concluding in Section 7.

2 Automatic Evaluation Metrics

In this section, we describe BLEU, and the three metrics which achieved higher correlation results than BLEU in the recent ACL-07 MT workshop.

2.1 BLEU

BLEU (Papineni et al., 2002) is essentially a precision-based metric and is currently the standard metric for automatic evaluation of MT performance. To score a system translation, BLEU tabulates the number of n-gram matches of the system translation against one or more reference translations. Generally, more n-gram matches result in a higher BLEU score.

When determining the matches to calculate precision, BLEU uses a *modified*, or *clipped* n-gram precision. With this, an n-gram (from both the system and reference translation) is considered to be exhausted or used after participating in a match. Hence, each system n-gram is "clipped" by the maximum number of times it appears in any reference translation.

To prevent short system translations from receiving too high a score and to compensate for its lack of a recall component, BLEU incorporates a brevity penalty. This penalizes the score of a system if the length of its entire translation output is shorter than the length of the reference text.

2.2 Semantic Roles

(Gimenez and Marquez, 2007) proposed using deeper linguistic information to evaluate MT performance. For evaluation in the ACL-07 MT workshop, the authors used the metric which they termed as SR-O_r-*[1]. This metric first counts the number of lexical overlaps SR-O_r-t for all the different semantic roles t that are found in the system and reference translation sentence. A uniform average of the counts is then taken as the score for the sentence pair. In their work, the different semantic roles t they considered include the various core and adjunct arguments as defined in the PropBank project (Palmer et al., 2005). For instance, SR-O_r-$A0$ refers to the number of lexical overlaps between the $A0$ arguments. To extract semantic roles from a sentence, several processes such as lemmatization, part-of-speech tagging, base phrase chunking, named entity tagging, and finally semantic role tagging need to be performed.

2.3 ParaEval

The ParaEval metric (Zhou et al., 2006) uses a large collection of paraphrases, automatically extracted from parallel corpora, to evaluate MT performance. To compare a pair of sentences, ParaEval first locates paraphrase matches between the two

[1] Verified through personal communication as this is not evident in their paper.

sentences. Then, unigram matching is performed on the remaining words that are not matched using paraphrases. Based on the matches, ParaEval will then elect to use either unigram precision or unigram recall as its score for the sentence pair. In the ACL-07 MT workshop, ParaEval based on recall (ParaEval-recall) achieves good correlation with human judgements.

2.4 METEOR

Given a pair of strings to compare (a system translation and a reference translation), METEOR (Banerjee and Lavie, 2005) first creates a word alignment between the two strings. Based on the number of word or unigram matches and the amount of string fragmentation represented by the alignment, METEOR calculates a score for the pair of strings.

In aligning the unigrams, each unigram in one string is mapped, or linked, to at most one unigram in the other string. These word alignments are created incrementally through a series of stages, where each stage only adds alignments between unigrams which have not been matched in previous stages. At each stage, if there are multiple different alignments, then the alignment with the most number of mappings is selected. If there is a tie, then the alignment with the least number of unigram mapping crosses is selected.

The three stages of “exact”, “porter stem”, and “WN synonymy” are usually applied in sequence to create alignments. The “exact” stage maps unigrams if they have the same surface form. The “porter stem” stage then considers the remaining unmapped unigrams and maps them if they are the same after applying the Porter stemmer. Finally, the “WN synonymy” stage considers all remaining unigrams and maps two unigrams if they are synonyms in the WordNet sense inventory (Miller, 1990).

Once the final alignment has been produced, unigram precision *P* (number of unigram matches *m* divided by the total number of system unigrams) and unigram recall *R* (*m* divided by the total number of reference unigrams) are calculated and combined into a single parameterized harmonic mean (Rijsbergen, 1979):

$$F_{mean} = \frac{P \cdot R}{\alpha P + (1-\alpha)R} \tag{1}$$

To account for longer matches and the amount of fragmentation represented by the alignment, METEOR groups the matched unigrams into as few *chunks* as possible and imposes a penalty based on the number of chunks. The METEOR score for a pair of sentences is:

$$score = \left[1 - \gamma \left(\frac{\text{no. of } chunks}{m}\right)^{\beta}\right] F_{mean}$$

where $\gamma \left(\frac{\text{no. of } chunks}{m}\right)^{\beta}$ represents the fragmentation penalty of the alignment. Note that METEOR consists of three parameters that need to be optimized based on experimentation: α, β, and γ.

3 Metric Design Considerations

We first review some aspects of existing metrics and highlight issues that should be considered when designing an MT evaluation metric.

- **Intuitive interpretation**: To compensate for the lack of recall, BLEU incorporates a brevity penalty. This, however, prevents an intuitive interpretation of its scores. To address this, standard measures like precision and recall could be used, as in some previous research (Banerjee and Lavie, 2005; Melamed et al., 2003).

- **Allowing for variation**: BLEU only counts exact word matches. Languages, however, often allow a great deal of variety in vocabulary and in the ways concepts are expressed. Hence, using information such as synonyms or dependency relations could potentially address the issue better.

- **Matches should be weighted**: Current metrics either match, or don’t match a pair of items. We note, however, that matches between items (such as words, n-grams, etc.) should be weighted according to their *degree* of similarity.

4 The Maximum Similarity Metric

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

Given a pair of English sentences to be compared (a system translation against a reference translation), we perform tokenization[2], lemmatization using WordNet[3], and part-of-speech (POS) tagging with the MXPOST tagger (Ratnaparkhi, 1996). Next, we remove all non-alphanumeric tokens. Then, we match the unigrams in the system translation to the unigrams in the reference translation. Based on the matches, we calculate the recall and precision, which we then combine into a single F_{mean} unigram score using Equation 1. Similarly, we also match the bigrams and trigrams of the sentence pair and calculate their corresponding F_{mean} scores. To obtain a single similarity score $score_s$ for this sentence pair s, we simply average the three F_{mean} 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 average over all $|S|$ sentence pairs:

$$sim\text{-}score = \frac{1}{|S|}\sum_{s=1}^{|S|}\left[\frac{1}{N}\sum_{n=1}^{N} F_{mean_{s,n}}\right]$$

where in our experiments, we set N=3, representing calculation of unigram, bigram, and trigram scores. 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 average the scores obtained.

4.1 Using N-gram Information

In this subsection, we describe in detail how we match the n-grams of a system-reference sentence pair.

Lemma and POS match 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 count the number $match_{uni}$ of system-reference unigram pairs where both their lemma and POS-tag match. To find matching pairs, we proceed in a left-to-right fashion (in both strings). We first compare the first system unigram to the first reference unigram, then to the second reference unigram, and so on until we find a match. If there is a match, we increment $match_{uni}$ by 1 and remove this pair of system-reference unigrams from further consideration (removed items will not be matched again subsequently). Then, we move on to the second system unigram and try to match it against the reference unigrams, once again proceeding in a left-to-right fashion. We continue this process until we reach the last system unigram.

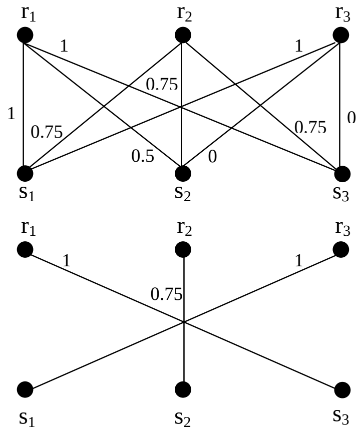

Figure 1: Bipartite matching.

To determine the number $match_{bi}$ of bigram matches, a system bigram $(l_{s_i}p_{s_i}, l_{s_{i+1}}p_{s_{i+1}})$ matches a reference bigram $(l_{r_i}p_{r_i}, l_{r_{i+1}}p_{r_{i+1}})$ if $l_{s_i} = l_{r_i}$, $p_{s_i} = p_{r_i}$, $l_{s_{i+1}} = l_{r_{i+1}}$, and $p_{s_{i+1}} = p_{r_{i+1}}$. For trigrams, we similarly determine $match_{tri}$ by counting the number of trigram matches.

Lemma match For the remaining set of n-grams 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 $(l_{r_i}p_{r_i})$ if $l_{s_i} = l_{r_i}$. In the case of bigrams, the matching conditions are $l_{s_i} = l_{r_i}$ and $l_{s_{i+1}} = l_{r_{i+1}}$. The conditions for trigrams are similar. Once again, we find matches in a left-to-right fashion. We add the number of unigram, bigram, and trigram matches found during this phase to $match_{uni}$, $match_{bi}$, and $match_{tri}$ respectively.

Bipartite graph matching For the remaining n-grams that are not matched so far, we try to match them by constructing bipartite graphs. During this phase, we will construct three bipartite graphs, one

[2] http://www.cis.upenn.edu/ treebank/tokenizer.sed

[3] http://wordnet.princeton.edu/man/morph.3WN

each for the remaining set of unigrams, bigrams, and trigrams.

Using bigrams to illustrate, we construct a weighted complete bipartite graph, where each edge e connecting a pair of system-reference bigrams has a weight $w(e)$, indicating the degree of similarity between the bigrams connected. Note that, without loss of generality, if the number of system nodes and reference nodes (bigrams) 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.

In an *n*-gram bipartite graph, the similarity score, or the weight $w(e)$ of the edge e connecting a system n-gram $(l_{s_1}p_{s_1}, \ldots, l_{s_n}p_{s_n})$ and a reference n-gram $(l_{r_1}p_{r_1}, \ldots, l_{r_n}p_{r_n})$ is calculated as follows:

$$\begin{aligned} S_i &= \frac{I(p_{s_i}, p_{r_i}) + Syn(l_{s_i}, l_{r_i})}{2} \\ w(e) &= \frac{1}{n}\sum_{i=1}^{n} S_i \end{aligned}$$

where $I(p_{s_i}, p_{r_i})$ evaluates to 1 if $p_{s_i} = p_{r_i}$, and 0 otherwise. The function $Syn(l_{s_i}, l_{r_i})$ checks whether l_{s_i} is a synonym of l_{r_i}. To determine this, we first obtain the set $WN_{syn}(l_{s_i})$ of WordNet synonyms for l_{s_i} and the set $WN_{syn}(l_{r_i})$ of WordNet synonyms for l_{r_i}. Then,

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

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. Further, if we are comparing bigrams or trigrams, we impose an additional condition: $S_i \neq 0$, for $1 \leq i \leq n$, 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.

In the top half of Figure 1, we show an example of a complete bipartite graph, constructed for a set of three system bigrams (s_1, s_2, s_3) and three reference bigrams (r_1, r_2, r_3), and the weight of the connecting edge between two bigrams represents their degree of similarity.

Next, we aim to find a *maximum* weight matching (or alignment) between the bigrams such that each system (reference) bigram is connected to exactly one reference (system) bigram. 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 1 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 bigram bipartite graph, we sum up the weights of the edges to obtain the weight of the matching M: $w(M) = \sum_{e \in M} w(e)$, and add $w(M)$ to $match_{bi}$. From the unigram and trigram bipartite graphs, we similarly calculate their respective $w(M)$ and add to the corresponding $match_{uni}$ and $match_{tri}$.

Based on $match_{uni}$, $match_{bi}$, and $match_{tri}$, we calculate their corresponding precision P and recall R, from which we obtain their respective F_{mean} scores via Equation 1. Using bigrams for illustration, we calculate its P and R as:

$$\begin{aligned} P &= \frac{match_{bi}}{\text{no. of bigrams in system translation}} \\ R &= \frac{match_{bi}}{\text{no. of bigrams in reference translation}} \end{aligned}$$

4.2 Dependency Relations

Besides matching a pair of system-reference sentences based on the surface form of words, previous work such as (Gimenez and Marquez, 2007) and (Rajman and Hartley, 2002) had shown that deeper linguistic knowledge such as semantic roles and syntax can be usefully exploited.

In the previous subsection, we describe our method of using bipartite graphs for matching of n-grams found in a sentence pair. This use of bipartite graphs, however, is a very general framework to obtain an optimal alignment of the corresponding “information items” contained within a sentence pair. Hence, besides matching based on n-gram strings, we can also match other “information items”, such as dependency relations.

Metric	Adequacy	Fluency	Rank	Constituent	Average
MaxSim$_{n+d}$	0.780	0.827	0.875	0.760	0.811
MaxSim$_{n}$	**0.804**	**0.845**	**0.893**	0.766	**0.827**
Semantic-role	0.774	0.839	0.804	0.742	0.790
ParaEval-recall	0.712	0.742	0.769	**0.798**	0.755
METEOR	0.701	0.719	0.746	0.670	0.709
BLEU	0.690	0.722	0.672	0.603	0.672

Table 1: Overall correlations on the Europarl and News Commentary datasets. The "Semantic-role overlap" metric is abbreviated as "Semantic-role". Note that each figure above represents 6 translation tasks: the Europarl and News Commentary datasets each with 3 language pairs (German-English, Spanish-English, French-English).

In our work, we train the MSTParser[4] (McDonald et al., 2005) on the Penn Treebank Wall Street Journal (WSJ) corpus, and use it to extract dependency relations from a sentence. Currently, we focus on extracting only two relations: *subject* and *object*. For each relation (ch, dp, pa) extracted, we note the child lemma ch of the relation (often a noun), the relation type dp (either *subject* or *object*), and the parent lemma pa of the relation (often a verb). Then, using the system relations and reference relations extracted from a system-reference sentence pair, we similarly construct a bipartite graph, where each node is a relation (ch, dp, pa). We define the weight $w(e)$ of an edge e between a system relation (ch_s, dp_s, pa_s) and a reference relation (ch_r, dp_r, pa_r) as follows:

$$\frac{Syn(ch_s, ch_r) + I(dp_s, dp_r) + Syn(pa_s, pa_r)}{3}$$

where functions I and Syn are defined as in the previous subsection. Also, $w(e)$ is non-zero only if $dp_s = dp_r$. After solving for the maximum weight matching M, we divide $w(M)$ by the number of system relations extracted to obtain a precision score P, and divide $w(M)$ by the number of reference relations extracted to obtain a recall score R. P and R are then similarly combined into a F_{mean} score for the sentence pair. To compute the similarity score when incorporating dependency relations, we average the F_{mean} scores for unigrams, bigrams, trigrams, and dependency relations.

5 Results

To evaluate our metric, we conduct experiments on datasets from the ACL-07 MT workshop and NIST MT 2003 evaluation exercise.

Europarl					
Metric	Adq	Flu	Rank	Con	Avg
MaxSim$_{n+d}$	0.749	0.786	**0.857**	0.651	**0.761**
MaxSim$_{n}$	0.749	0.786	**0.857**	0.651	**0.761**
Semantic-role	**0.815**	**0.854**	0.759	0.612	0.760
ParaEval-recall	0.701	0.708	0.737	**0.772**	0.730
METEOR	0.726	0.741	0.770	0.558	0.699
BLEU	0.803	0.822	0.699	0.512	0.709

Table 2: Correlations on the Europarl dataset. Adq=Adequacy, Flu=Fluency, Con=Constituent, and Avg=Average.

News Commentary					
Metric	Adq	Flu	Rank	Con	Avg
MaxSim$_{n+d}$	0.812	0.869	0.893	0.869	0.861
MaxSim$_{n}$	**0.860**	**0.905**	**0.929**	**0.881**	**0.894**
Semantic-role	0.734	0.824	0.848	0.871	0.819
ParaEval-recall	0.722	0.777	0.800	0.824	0.781
METEOR	0.677	0.698	0.721	0.782	0.720
BLEU	0.577	0.622	0.646	0.693	0.635

Table 3: Correlations on the News Commentary dataset.

5.1 ACL-07 MT Workshop

The ACL-07 MT workshop evaluated the translation quality of MT systems on various translation tasks, and also measured the correlation (with human judgement) of 11 automatic MT evaluation metrics. The workshop used a Europarl dataset and a News Commentary dataset, where each dataset consisted of English sentences (2,000 English sentences for Europarl and 2,007 English sentences for News Commentary) and their translations in various languages. As part of the workshop, correlations of the automatic metrics were measured for the tasks

[4] Available at: http://sourceforge.net/projects/mstparser

of translating German, Spanish, and French into English. Hence, we will similarly measure the correlation of MAXSIM on these tasks.

5.1.1 Evaluation Criteria

For human evaluation of the MT submissions, four different criteria were used in the workshop: **Adequacy** (how much of the original meaning is expressed in a system translation), **Fluency** (the translation's fluency), **Rank** (different translations of a single source sentence are compared and ranked from best to worst), and **Constituent** (some constituents from the parse tree of the source sentence are translated, and human judges have to rank these translations).

During the workshop, Kappa values measured for inter- and intra-annotator agreement for *rank* and *constituent* are substantially higher than those for *adequacy* and *fluency*, indicating that *rank* and *constituent* are more reliable criteria for MT evaluation.

5.1.2 Correlation Results

We follow the ACL-07 MT workshop process of converting the raw scores assigned by an automatic metric to ranks and then using the Spearman's rank correlation coefficient to measure correlation.

During the workshop, only three automatic metrics (Semantic-role overlap, ParaEval-recall, and METEOR) achieve higher correlation than BLEU. We gather the correlation results of these metrics from the workshop paper (Callison-Burch et al., 2007), and show in Table 1 the overall correlations of these metrics over the Europarl and News Commentary datasets. In the table, MAXSIM$_n$ represents using only n-gram information (Section 4.1) for our metric, while MAXSIM$_{n+d}$ represents using both n-gram and dependency information. We also show the breakdown of the correlation results into the Europarl dataset (Table 2) and the News Commentary dataset (Table 3). In all our results for MAXSIM in this paper, we follow METEOR and use α=0.9 (weighing recall more than precision) in our calculation of F_{mean} via Equation 1, unless otherwise stated.

The results in Table 1 show that MAXSIM$_n$ and MAXSIM$_{n+d}$ achieve overall average (over the four criteria) correlations of 0.827 and 0.811 respectively. Note that these results are substantially higher than BLEU, and in particular higher than the *best* performing *Semantic-role overlap* metric in the ACL-07 MT workshop. Also, Semantic-role overlap requires more processing steps (such as base phrase chunking, named entity tagging, etc.) than MAXSIM. For future work, we could experiment with incorporating semantic-role information into our current framework. We note that the ParaEval-recall metric achieves higher correlation on the *constituent* criterion, which might be related to the fact that both ParaEval-recall and the *constituent* criterion are based on phrases: ParaEval-recall tries to match phrases, and the *constituent* criterion is based on judging translations of phrases.

Metric	Adq	Flu	Avg
MAXSIM$_{n+d}$	0.943	0.886	0.915
MAXSIM$_n$	0.829	0.771	0.800
METEOR (optimized)	1.000	0.943	0.972
METEOR	0.943	0.886	0.915
BLEU	0.657	0.543	0.600

Table 4: Correlations on the NIST MT 2003 dataset.

5.2 NIST MT 2003 Dataset

We also conduct experiments on the test data (LDC2006T04) of NIST MT 2003 Chinese-English translation task. For this dataset, human judgements are available on *adequacy* and *fluency* for six system submissions, and there are four English reference translation texts.

Since implementations of the BLEU and METEOR metrics are publicly available, we score the system submissions using BLEU (version 11b with its default settings), METEOR, and MAXSIM, showing the resulting correlations in Table 4. For METEOR, when used with its originally proposed parameter values of (α=0.9, β=3.0, γ=0.5), which the METEOR researchers mentioned were based on some early experimental work (Banerjee and Lavie, 2005), we obtain an average correlation value of 0.915, as shown in the row "METEOR". In the recent work of (Lavie and Agarwal, 2007), the values of these parameters were tuned to be (α=0.81, β=0.83, γ=0.28), based on experiments on the NIST 2003 and 2004 Arabic-English evaluation datasets. When METEOR was run with these new parameter values, it returned an average correlation value of

0.972, as shown in the row "METEOR (optimized)".

MAXSIM using only n-gram information (MAXSIM$_n$) gives an average correlation value of 0.800, while adding dependency information (MAXSIM$_{n+d}$) improves the correlation value to 0.915. Note that so far, the parameters of MAXSIM are not optimized and we simply perform uniform averaging of the different n-grams and dependency scores. Under this setting, the correlation achieved by MAXSIM is comparable to that achieved by METEOR.

6 Future Work

In our current work, the parameters of MAXSIM are as yet un-optimized. We found that by setting α=0.7, MAXSIM$_{n+d}$ could achieve a correlation of 0.972 on the NIST MT 2003 dataset. Also, we have barely exploited the potential of weighted similarity matching. Possible future directions include adding semantic role information, using the distance between item pairs based on the token position within each sentence as additional weighting consideration, etc. Also, we have seen that dependency relations help to improve correlation on the NIST dataset, but not on the ACL-07 MT workshop datasets. Since the accuracy of dependency parsers is not perfect, a possible future work is to identify when best to incorporate such syntactic information.

7 Conclusion

In this paper, we present MAXSIM, a new automatic MT evaluation metric that computes a similarity score between corresponding items across a sentence pair, and uses a bipartite graph to obtain an optimal matching between item pairs. This general framework allows us to use arbitrary similarity functions between items, and to incorporate different information in our comparison. When evaluated for correlation with human judgements, MAXSIM achieves superior results when compared to current automatic MT evaluation metrics.

References

S. Banerjee and A. Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization, ACL05*, pages 65–72.

C. Callison-Burch, C. Fordyce, P. Koehn, C. Monz, and J. Schroeder. 2007. (meta-) evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation, ACL07*, pages 136–158.

J. Gimenez and L. Marquez. 2007. Linguistic features for automatic evaluation of heterogenous MT systems. In *Proceedings of the Second Workshop on Statistical Machine Translation, ACL07*, pages 256–264.

H. W. Kuhn. 1955. The hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2(1):83–97.

A. Lavie and A. Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation, ACL07*, pages 228–231.

R. McDonald, K. Crammer, and F. Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of ACL05*, pages 91–98.

I. D. Melamed, R. Green, and J. P. Turian. 2003. Precision and recall of machine translation. In *Proceedings of HLT-NAACL03*, pages 61–63.

G. A. Miller. 1990. WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–312.

J. Munkres. 1957. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38.

M. Palmer, D. Gildea, and P. Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of ACL02*, pages 311–318.

M. Rajman and A. Hartley. 2002. Automatic ranking of MT systems. In *Proceedings of LREC02*, pages 1247–1253.

A. Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of EMNLP96*, pages 133–142.

C. Rijsbergen. 1979. *Information Retrieval*. Butterworths, London, UK, 2nd edition.

B. Taskar, S. Lacoste-Julien, and D. Klein. 2005. A discriminative matching approach to word alignment. In *Proceedings of HLT/EMNLP05*, pages 73–80.

L. Zhou, C. Y. Lin, and E. Hovy. 2006. Re-evaluating machine translation results with paraphrase support. In *Proceedings of EMNLP06*, pages 77–84.