# Using Collocations to Assess MT Quality

Benjamin Han and Alon Lavie

Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
benhdj@cs.cmu.edu

May 8, 2005

**Abstract**

Conventional metrics for Machine Translation evaluation have focused on using *n-gram* similarity between a reference translation and a system translation as an indication of the system quality. A simple n-gram model however cannot capture long-distance dependency, and the requirement of a reference translation has prevented the use of these metrics at the decoding stage. In this paper we propose a set of collocation-based metrics to address these problems. A series of experiments has shown that these metrics are capable of distinguishing between human translations and system translations, and can produce system rankings comparable to the other metrics without requiring a reference translation.

## 1 Introduction

Although designing an effective evaluation metric for machine translation (MT) has long been a highly sought objective [14, 1, 12, 9], automatic methods had not been widely deployed in the field until the recent years. Among them IBM's BLEU metric [7], developed based on the NIST metric [3], has been extensively used in the DARPA-sponsored TIDES MT program. Both methods compute scores based on the number of *n-grams* shared between a machine-generated translation and a reference translation, with an emphasis on the system's n-gram *precision*. This emphasis has prompted the development of a more *recall*-centric metric METEOR [8], which has been shown to be more correlated to the human judgments on system performance.

Despite the differences in their emphases, NIST, BLEU and METEOR are all based on the n-gram similarity between a generated translation and a reference translation. The adoption of n-gram models not only poses a major problem in capturing the phenomena of long-distance dependency that usually manifests in natural language [5], but could potentially give biased results due to the wide-spread use of such models in MT systems.

In this paper we propose using *collocations* as a way to assess MT quality. To motivate our approach, consider the following Chinese-to-English translations[1]:

[1]Taken from the actual data of the 2002 Tides MT evaluation.

**Reference translation**: "*The average price for 30 main industrial stocks of Dow Jones was down 97 . 15 points throughout the day...*"
**System output**: "*Dow Jones Industrial Average all dropped 97 dot worlds on average...*"

Clearly the system output erred on the use of the word *dot*, due to the fact that in Chinese a single character can mean both a *dot* and a *point*. Had we obtained a ranked list of collocations from an English corpus, and observed that the word *Jones* and *points* co-occur more often than *Jones* and *dot*, we would have penalized such a translation more than the ones that used the correct word *points*. Note that this penalization is not directly possible in a simple n-gram model, as *Jones* and *points* usually do not occur in adjacent positions.

A second observation from the example is that the system output can be penalized even without the presence of a reference translation. The list of collocations is obtained from an independent corpus, and can be prepared in advance. This makes a collocation-based metric not only useful in ranking system performances, but could also be beneficial in ranking outputs from a single system (decoding).

To establish the feasibility of using a collocation-based metric, we set out to fulfill the following goals: (i) we want to distinguish reference translations from system output reliably using the metric; (ii) we want to produce a system ranking that is highly similar to human judgments. We will first describe methods of scoring collocations, and then use the metric to compute sentence scores (i.e., average collocation scores) and system scores (i.e., average sentence scores).

The rest of the paper is organized as follows. In Sec. 2 we first give our definition of collocations and the four metrics for scoring them. Sec. 3 then describes the four variations in computing a sentence-level score based on the collocation metrics. Sec. 4 shows the results of our experiments with respect to the goals we outlined above. Finally Sec. 5 concludes this paper, and offers a list of future work.

## 2 Finding Collocations

A collocation is simply a pair of co-occurring words in a sentence. Before collection collocations from a sentence, we first preprocess it as follows:

1. Tokenize and part-of-speech-tag the sentence;
2. Filter out all words that are not nouns, verbs, adjectives and adverbs;
3. Use WordNet [4] to canonicalize the remaining words (we call them *content words*) based on their part-of-speech;
4. Remove duplicate words so each word is unique.

For example, the sentence "*Mussa's one-day trip coincides with Sudanese Foreign Minister Mustafa Uthman Ismail's visit who arrived in Tripoli today.*" is preprocessed into "*mussa one-day trip coincide sudanese foreign minister mustafa uthman ismail visit arrived tripoli today*" and every possible pair is collected as a collocation.

We then assign a score to each collocation using one of the four standard metrics: the *dice metric*, the $t$ score, the $\chi^2$ score, and the *likelihood ratio*. In the beginning we used the WSJ

| Collocation | Score |
|---|---|
| *zeitung zuercher* | 1.0 |
| *ymca ywca* | 1.0 |
| *yankee yastrzemski* | 1.0 |
| *yaniv zvi* | 1.0 |
| ... | ... |
| *hong kong* | 0.974504 |
| *mixte navigation* | 0.974359 |
| *freddie mac* | 0.967742 |
| *fulton prebon* | 0.965517 |
| *du pont* | 0.961039 |
| ... | ... |

Table 1: Example collocations obtained using the dice metric over the WSJ sections of Treebank

sections of Treebank [10] for collecting collocations. The corpus contains 49,722 sentences, with 32,411 unique content words and 2,459,065 collocations (about 60 MB in terms of the disk space). The example collocations shown in the following sub-sections were all obtained from the Treebank corpora. In the experiments described in Sec. 3 and 4, however, we used a much larger corpus - one month worth of New York Times from the English Gigaword corpus [6][2]. The corpus contains 629,164 sentences, with 44,713 unique content words and 11,307,512 collocations (about 780 MB for each of the four generated collocation lists).

We now give short descriptions for each collocation metric in the following sub-sections.

## 2.1 Dice Metric

One commonly used metric in identifying strength of word associations in Information Retrieval has been the dice metric [13]. It is formulated as

$$s = \frac{2 \times c_{12}}{c_1 + c_2}$$

where $c_1$ and $c_2$ are the frequencies of two words in a training corpus, and $c_{12}$ is their joint frequency. Table 1 shows some of the collocations found in the WSJ sections of Treebank[3]. Note that many top collocations (about 0.28% of all collocations) share score 1.0 - they are special cases when $c_1 = c_2 = c_{12} = 1$.

## 2.2 $t$ Score

Collocation-finding can be considered as a null hypothesis test, where the independence assumption of the occurrence of any two words is the null hypothesis. If we further assume a binomial distribution for collocations, we can compute a $t$ score (s) for a collocation as follows

[2] January 1997.

[3] The ordering between the two words in a collocation is *not* significant.

| Collocation | Score |
|---|---|
| *do n't* | 35.729019 |
| *new york* | 32.843865 |
| *cent share* | 22.145876 |
| *earlier year* | 21.393558 |
| *exchange stock* | 20.737076 |
| ... | ... |
| *jones point* | 7.528200 |
| ... | ... |
| *airline transaction* | 2.576339 |
| *aid congress* | 2.576338 |
| ... | ... |
| *bay traffic* | 2.576311 |
| ... | ... |

Table 2: Example collocations obtained using the $t$ score over the WSJ sections of Treebank

| | $w_1 = W_1$ | $w_1 \neq W_1$ |
|---|---|---|
| $w_2 = W_2$ | $O_{11}$ | $O_{12}$ |
| $w_2 \neq W_2$ | $O_{21}$ | $O_{22}$ |

Figure 1: 2-by-2 contingency table for $\chi^2$ tests

$$\begin{aligned} s &= \frac{\bar{x} - \mu}{\sqrt{\sigma^2/N}} \\ &\approx \frac{p_{12} - p_1 p_2}{\sqrt{p_{12}/N}} \end{aligned}$$

where $p_1$ and $p_2$ are word probabilities, $p_{12}$ is collocation probability, and $N$ is the sample size. When we compute $s > 2.576$ for a collocation, we have 99.5% confidence that the collocation is real. Table 2 shows some of the collocations found in the WSJ sections of Treebank. Note that the top-most collocations are just typical bigrams, but as the list goes down to near the critical value 2.576, more and more collocations capture longer-distance dependencies. Also note the strong collocation "*jones point*" (see the example in Sec. 1); the collocation "*dot jones*" on the other hand never occurred.

## 2.3 $\chi^2$ Score

Testing dependency of two words can be done using $\chi^2$ scores without assuming an underlying distribution of the collocations. First we build a 2-by-2 contingency table shown in Fig. 1. The score can then be formulated as

| Collocation | Score |
|---|---|
| *zeitung zuercher* | 49722.000000 |
| *ymca ywca* | 49722.000000 |
| *yankee yastrzemski* | 49722.000000 |
| *yaniv zvi* | 49722.000000 |
| ... | ... |
| *hong kong* | 47241.054744 |
| *mixte navigation* | 47233.046901 |
| *freddie mac* | 46611.560122 |
| *fulton prebon* | 46406.266423 |
| *du pont* | 45990.073101 |
| ... | ... |

Table 3: Example collocations obtained using the $\chi^2$ score over the WSJ sections of Treebank

$$s = \frac{N(O_{11}O_{22} - O_{12}O_{21})^2}{(O_{11} + O12)(O_{11} + O21)(O_{12} + O22)(O_{21} + O22)}.$$

Table 3 shows some of the collocations found in the WSJ sections of Treebank. Note that this ranked list is very similar to the list computed using the dice metric.

### 2.4 Likelihood Ratio

Another way to test word dependency is to compute the likelihood ratio of the two hypotheses: (i) $H_1$ (independence hypothesis): $p(w_2|w_1) = p = p(w_2|\neg w_1)$ and (ii) $H_2$: $p(w_2|w_1) = p_1 \neq p_2 = p(w_2|\neg w_1)$. Assuming a binomial distribution, we have

$$\begin{aligned} s &= -2\log\frac{L(H_1)}{L(H_2)} \\ &= -2(\log L(c_{12}, c_1, p) + \log L(c_2 - c_{12}, N - c_1, p) - \log L(c_{12}, c_1, p_1) - \log L(c_2 - c_{12}, N - c_1, p_2)) \end{aligned}$$

Table 4 shows some of the collocations found in the WSJ sections of Treebank. Note that this ranked list is similar to the list computed using the $t$ score.

## 3 Computing Sentence and System Scores

After obtaining a scored list of collocations using one of the four metrics outlined above, given a translated sentence, we can simply find all of the collocations appearing inside the sentence, and compute the average collocation score as the *sentence score*. For example, given a translated sentence "$A\ B\ B\ C$", if only collocation "$A\ C$" and "$B\ C$" have been collected over a training corpus with score 1.0 and 2.0, then the sentence score is $(1.0 + 2.0)/2 = 1.5$. Note that due to the preprocessing procedure, only one copy of $B$ would be admitted in the calculation.

| Collocation | Score |
|---|---|
| *new york* | 10133.2241008 |
| *do n't* | 8124.73484344 |
| *street wall* | 4868.5440851 |
| *chief officer* | 4427.38280735 |
| *francisco san* | 4271.57303174 |
| *dow jones* | 4229.75863621 |
| ... | ... |
| *company statement* | 86.8237568468 |
| *democratic republican* | 86.8038966121 |
| *average banks* | 86.8003659115 |
| *book write* | 86.7912248281 |
| ... | ... |

Table 4: Example collocations obtained using the $t$ score over the WSJ sections of Treebank

To rank MT systems, we can compute the average sentence score over every sentence a system translated as its *system score*, and rank them accordingly. This will fulfill both of the goals we outlined in Sec. 1.

There is however a potential problem with the simple approach outlined above. Take the example sentence used in the beginning of Sec. 2:

> "*Mussa's one-day trip coincides with Sudanese Foreign Minister Mustafa Uthman Ismail's visit who arrived in Tripoli today.*"

If we use the simple method to compute the sentence score, we would take the collocation "*minister today*" into consideration. However since the prepositional phrase "*today*" can be attached to a wide range of verb phrases, the fact that the word co-occurred with "*minister*" should not affect our judgment as to whether the translation looks more or less plausible. This intuition can be enforced by using the dependency structure [11] of the sentence:


Note that there is no dependency between the word "*minister*" and "*today*" (no branch connecting the two). We can simply stipulate that for a collocation to be considered in computing a sentence score, the two words must be connected in the dependency structure of the sentence.

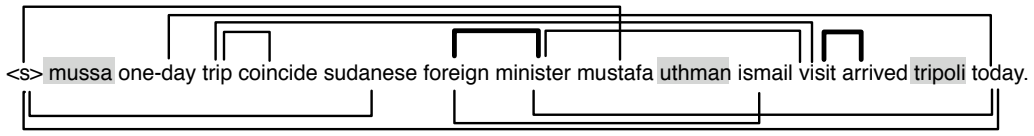
Unfortunately in practice such structures are usually not available. Even if we are indulged with the luxury of dependency parsing on the translations, the resulting parses would be unreliable due to the errors made in the MT systems.

In the following sub-sections we describe three successively more elaborated methods to "fake" the dependency structures. It should be emphasized that we are not trying to produce a 100% accurate dependency structure; our overall goal here is to constrain the sentence score computation by using grammatically motivated structures. In Sec. 4 these variations will each be tested out.

### 3.1 Computing Sentence Scores using Maximum Spanning Trees (MST)

A dependency structure is basically a tree, with words serving as vertices and dependencies as tree branches. The most straightforward way to use such structures in computing sentence scores is to enumerate all possible trees for a sentence and calculate the average collocation scores (i.e., considering only the collocations where one word depends on the other). However a corollary of the *matrix tree theorem* states that a complete graph with $n$ vertices contains $n^{n-2}$ spanning trees, which makes the brute-force approach clearly infeasible. A second proposal would be to compute the maximum spanning tree (MST) out of a sentence, and use it as a representative tree: we first construct a graph where the content words are vertices, and each collocation that can be found in our pre-computed list forms an edge weighted by the collocation score; we then use Kruskal's algorithm to find the MST in polynomial-time [2]. Once the MST is decided, we can then compute the average collocation score, i.e., sentence score, by calculating the average edge weight.

Taking the same example sentence used earlier in this section, we can obtain the following MST, with a sentence score 3.019 (using $t$ score):




Note that the shaded words are disconnected since they did not show up in our collocation list (pre-computed using the English Gigaword corpus; see Sec. 2). The bold branches are the correct ones (compare with the correct structure given above): in this case we have two.

### 3.2 MST with No Crossing Branches (MST-NCB)

Comparing the correct dependency structure and the MST obtained above, one obvious flaw of our tree is that it has *crossing branches*; this is because we did not take word ordering into account. If we modify Kruskal's algorithm to discard crossing branches, the MST of the example sentence would change into the following:


In this case we are gaining one additional correct branch, with sentence score increasing to 3.764. Note that more words become disconnected because of the discarded branches.

### 3.3 MST-NCB with an Initial Branch (MST-NCB2)

Looking at the MST-NCB structure above, there is one mistake we can easily fix. Usually the root node `<s>` is connected to the first verb scanning from left to right (the main predicate). With this modification to Kruskal's algorithm, we have

Again we are gaining one additional correct branch, but the sentence score is decreased to 3.356. Note obviously this modification does not always work; e.g., it would not work with sentence "*The book I bought last week was stolen*".

# 4 Experiments and Results

To demonstrate that both goals outlined in Sec. 1 can be fulfilled, we first computed a list of scored collocations using a portion of the English Gigaword corpus and the four metrics described in Sec. 2. We then calculated the system scores over the data of 2002 and 2003 Tides MT evaluation, using the four variations outlined in Sec. 3. In this section we will first report on the task of distinguishing human and machine translations, and then compare the collocation-based system rankings with human judgments and other conventional metrics. Table 5 gives some basic statistics of our testing data.

| | **# of Humans** | **# of Systems** | **# of Sentences** |
|---|---|---|---|
| **2002 Arabic** | 4 | 3 | 728 |
| **2002 Chinese** | 4 | 7 | 878 |
| **2003 Arabic** | 4 | 6 | 338 |
| **2003 Chinese** | 4 | 11 | 919 |

Table 5: Basic statistics of our testing data

## 4.1 Distinguishing Humans from Systems

To test if our metric can successfully tell the difference between the reference translations (humans) and the system translations (systems), we calculated the average system scores for both sides. A *separation ratio* is then computed as follows:

$$\text{Separation ratio} = (\text{Human average} - \text{System average})/|\text{Human average}|$$

A positive separation ratio thus indicates the successful completion of this task. In fact the higher the ratio is, the better distinction a particular method achieves.

We now report the results along two dimensions. The first dimension is along the four collocation metrics described in Sec. 2. For each metric we only report on the sentence scoring method that achieved on average the best separation ratios. Table 6, 7, 8 and 9 give the results (shaded columns represent the best performing metric). Overall speaking, we confirmed that the collocation-based metric is capable of telling human translations and machine translations apart. The Chinese data seems to favor more the dice metric and the $\chi^2$ score, while the Arabic data tends to favor more the $t$ score and the likelihood ratio. The results also suggest that telling apart humans and machines in translating Chinese is much more difficult than doing the same for Arabic.

The other dimension of our report is along the four variations in computing sentence scores, outlined in Sec. 3. Table 10, 11, 12 and 13 give the results (shaded columns represent the best performing variation). Overall speaking for 2002 the Simple and MST variations are the two best performing methods, but for 2003 the MST-NCB and MST-NCB2 become the winners. Our

| | Dice Metric | $t$ Score | $\chi^2$ Score | Likelihood Ratio |
|---|---|---|---|---|
| **Human Average** | 0.0620049 | 7.5624950 | 11009.499 | 559.36625 |
| **System Average** | 0.0482311 | 5.6388967 | 8420.9759 | 337.50552 |
| **Separation Ratio** | 22.21% | 25.44% | 23.51% | 39.66% |

Table 6: 2002 Arabic - Effect of different collocation metrics (using MST - the best performing one among the 4 variations)

| | Dice Metric | $t$ Score | $\chi^2$ Score | Likelihood Ratio |
|---|---|---|---|---|
| **Human Average** | 0.0633512 | 7.9221359 | 11604.130 | 625.73902 |
| **System Average** | 0.0595995 | 7.5500239 | 11242.706 | 600.59049 |
| **Separation Ratio** | 5.92% | 4.70% | 3.11% | 4.02% |

Table 7: 2002 Chinese - Effect of different collocation metrics (using MST - the best performing one among the 4 variations)

| | Dice Metric | $t$ Score | $\chi^2$ Score | Likelihood Ratio |
|---|---|---|---|---|
| **Human Average** | 0.0636795 | 6.8804258 | 12029.167 | 574.46616 |
| **System Average** | 0.0532616 | 6.0165137 | 9782.2846 | 456.69555 |
| **Separation Ratio** | 16.36% | 12.56% | 18.68% | 20.50% |

Table 8: 2003 Arabic - Effect of different collocation metrics (using MST-NCB2 - the best performing one among the 4 variations)

| | Dice Metric | $t$ Score | $\chi^2$ Score | Likelihood Ratio |
|---|---|---|---|---|
| **Human Average** | 0.0728313 | 9.3319466 | 13288.081 | 802.48052 |
| **System Average** | 0.0655111 | 8.8165155 | 11835.198 | 762.27504 |
| **Separation Ratio** | 10.05% | 5.52% | 10.93% | 5.01% |

Table 9: 2003 Chinese - Effect of different collocation metrics (using MST - the best performing one among the 4 variations)

conjecture for this change is that due to the improvement of the systems over time, more elaborated approaches become more effective when picking collocations to compute sentence scores.

We conclude this sub-section with a discussion on the apparent disparity in the results between Arabic and Chinese. In both dimensions we reported above, the separation ratios achieved for the Chinese data are significantly lower than those achieved for the Arabic data (about 30 absolute percentage points for the 2002 data and 10 for the 2003; we even saw a negative ratio in the 2003 result). Our conjecture is that the discrepancy comes from the intrinsic difference between the two languages. More specifically, *segmentation* is a unique problem in Chinese - since there is no space separating the characters composing different constituents, most MT systems attempt to "segment" a Chinese sentence before translating it. This process could lead to translations with diverse sentence scores, both due to the sensitivity of word choice given different segmentations, and due to its influence to the grammatical structures of the translation.

## 4.2 Ranking MT Systems

Our second goal outlined in Sec. 1 is to produce performance rankings for MT systems that are *similar* to the human judgments. To evaluate the similarity between two ranked lists, we use the following simple procedure. Given two fully-ordered lists $l_1$ and $l_2$ of the same length, we first break each of them into a set of relational pairs $\mathcal{R}(l_1)$ and $\mathcal{R}(l_2)$. The similarity between the two lists is then computed as

$$\text{sim}(l_1, l_2) = \frac{|\mathcal{R}(l_1) \cap \mathcal{R}(l_2)|}{|\mathcal{R}(l_1)|}$$

For example, if $l_1$ is $a > b > c$ and $l_2$ is $b > c > a$, we have $\mathcal{R}(l_1) = \{(a,b),(a,c),(b,c)\}$, $\mathcal{R}(l_2) = \{(b,c),(b,a),(c,a)\}$, and $\text{sim}(l_1, l_2) = 1/3$.

We now report the similarity between the collocation-based ranked lists and the human judgments on the Chinese translations in Tides MT 2002 and 2003[4]. Table 14 shows the similarity scores of the lists produced by the 16 combinations of the methods (4 collocation metrics with 4 variations on computing sentence scores). The best similarity score 0.857143 was achieved using $t$ score with the MST sentence scoring method. In comparison, the rankings based on the BLEU metric over the same set of data has similarity score 0.8095238, and the rankings based on the NIST metric achieved similarity score 0.9047619. In short our collocation-based metric performed admirably between the BLEU metric and the NIST metric, but we do not require the presence of a reference translation. Another point worth noting is that MST seems to significantly boost the similarity scores across board.

For the 2003 Chinese data we have two sets of human judgments. Table 15 shows the similarity scores comparing the collocation-based ranked lists with the *adequacy* human judgments, and Table 16 gives the scores with respect to the *fluency* human judgments. Again the $t$ score achieved the highest similarity, but there is no difference shown among the family of the MST methods. Also note that the collocation-based metrics seem to agree more with the fluency judgments. For comparison the similarity scores achieved by the NIST metric, the BLEU metric and the METEOR metric are shown in Table 17.

[4]We were unable to obtain the data for Arabic at the time of experiments.

| | Simple | MST | MST-NCB | MST-NCB2 |
|---|---|---|---|---|
| **Human Average** | 119.76790 | 559.36625 | 549.90046 | 549.90046 |
| **System Average** | 84.589610 | 337.50552 | 334.29914 | 334.25972 |
| **Separation Ratio** | 29.37% | 39.66% | 39.21% | 39.21% |

Table 10: 2002 Arabic - Effect of different ways of computing sentence scores (using the likelihood ratio - the best performing one among the 4 metrics)

| | Simple | MST | MST-NCB | MST-NCB2 |
|---|---|---|---|---|
| **Human Average** | 0.0127268 | 0.0633512 | 0.0609119 | 0.0597488 |
| **System Average** | 0.0119374 | 0.0595995 | 0.0576491 | 0.0566402 |
| **Separation Ratio** | 6.20% | 5.92% | 5.36% | 5.20% |

Table 11: 2002 Chinese - Effect of different ways of computing sentence scores (using the dice metric - the best performing one among the 4 metrics)

| | Simple | MST | MST-NCB | MST-NCB2 |
|---|---|---|---|---|
| **Human Average** | 120.27843 | 579.96496 | 574.47475 | 574.46616 |
| **System Average** | 97.302655 | 464.98311 | 456.69555 | 456.69555 |
| **Separation Ratio** | 19.10% | 19.83% | 20.50% | 20.50% |

Table 12: 2003 Arabic - Effect of different ways of computing sentence scores (using the likelihood ratio - the best performing one among the 4 metrics)

| | Simple | MST | MST-NCB | MST-NCB2 |
|---|---|---|---|---|
| **Human Average** | 1027.9084 | 13288.081 | 13184.918 | 13108.019 |
| **System Average** | 1050.6777 | 11835.198 | 11756.706 | 11645.443 |
| **Separation Ratio** | -2.22% | 10.93% | 10.83% | 11.16% |

Table 13: 2003 Chinese - Effect of different ways of computing sentence scores (using the $\chi^2$ score - the best performing one among the 4 metrics)

# 5 Conclusions and Future Work

In this paper we first proposed a set of collocation-based metrics for MT evaluation. The metrics address the following problems from using the conventional metrics such as NIST, BLEU and METEOR: (i) a simple n-gram model cannot capture long-distance dependency frequently observed in natural language; (ii) wide-spread use of n-gram-based language models in MT systems could lead to a bias when using these metrics; and (iii) the requirement of a reference translation makes these metrics impossible to use at the decoding stage.

We then conducted a series of experiments to demonstrate the feasibility of using the collocation-based metrics to (i) distinguish between human translations and machine translations; and (ii) to produce system rankings of high fidelity. We also investigated the effects of using four different collocation metrics (the dice metric, the $t$ score, the $\chi^2$ score and the likelihood ratio), and the effects of using four variations in computing sentence scores (simple collocations, MST, MST-NCB and MST-NCB2). The results from these experiments proved to be promising.

In the future we would like to investigate ways of integrating these metrics into the decoding stage, and demonstrate their utility with empirical results. We would also like to revisit an assumption we have made when inducing dependency structures using collocation strengths; i.e., strong collocations are equivalent to dependencies. This will probably include an analysis on the correlation between the two notions over a structured corpus (such as Treebank). The insight thus gained can help us to further refine our collocation model; e.g., to incorporate additional information such as simple part-of-speech patterns and word distances.

| | Simple | MST | MST-NCB | MST-NCB2 |
|---|---|---|---|---|
| **Dice Metric** | 0.571429 | 0.714286 | 0.714286 | 0.714286 |
| $t$ **Score** | 0.666667 | 0.857143 | 0.809524 | 0.809524 |
| $\chi^2$ **Score** | 0.52381 | 0.666667 | 0.666667 | 0.666667 |
| **Likelihood Ratio** | 0.52381 | 0.714286 | 0.761905 | 0.761905 |

Table 14: 2002 Chinese - Similarity between the collocation-based ranked lists and the human judgments

| | Simple | MST | MST-NCB | MST-NCB2 |
|---|---|---|---|---|
| **Dice Metric** | 0.466667 | 0.466667 | 0.466667 | 0.466667 |
| $t$ **Score** | 0.4 | 0.6 | 0.6 | 0.6 |
| $\chi^2$ **Score** | 0.6 | 0.466667 | 0.466667 | 0.466667 |
| **Likelihood Ratio** | 0.4 | 0.533333 | 0.533333 | 0.533333 |

Table 15: 2003 Chinese - Similarity between the collocation-based ranked lists and the human judgments (*adequacy*)

| | Simple | MST | MST-NCB | MST-NCB2 |
|---|---|---|---|---|
| **Dice Metric** | 0.6 | 0.6 | 0.6 | 0.6 |
| $t$ **Score** | 0.533333 | 0.733333 | 0.733333 | 0.733333 |
| $\chi^2$ **Score** | 0.733333 | 0.6 | 0.6 | 0.6 |
| **Likelihood Ratio** | 0.533333 | 0.666667 | 0.666667 | 0.666667 |

Table 16: 2003 Chinese - Similarity between the collocation-based ranked lists and the human judgments (*fluency*)

| | Adequacy | Fluency |
|---|---|---|
| **NIST** | 0.8667 | 0.8667 |
| **BLEU** | 0.7333 | 0.8667 |
| **METEOR** | 0.8667 | 0.8667 |

Table 17: 2003 Chinese - Similarity scores achieved by the NIST metric, the BLEU metric and the METEOR metric

# References


[1] Y. Akiba, K. Imamura, and E. Sumita. Using multiple edit distances to automatically rank machine translation output. In *Proceedings of MT Summit VIII. Santiago de Compostela*, pages 15–20, Spain, 2001.

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. McGraw-Hill, 2nd edition, 2001.

[3] George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Second Conference on Human Language Technology (HLT-2002)*, pages 128–132, San Diego, CA, 2002.

[4] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, May 1998.

[5] Jianfeng Gao and Hisami Suzuki. Capturing long distance dependency for language modeling: an empirical study. In *Proceedings of the First International Joint Conference on Natural Language Processing (IJCNLP-04)*, Sanya City, Hainan Island, China, March 2004.

[6] David Graff. LDC English Gigaword Corpus, LDC2003T05. http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2003T05.

[7] Papineni Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, PA, July 2002.

[8] A. Lavie, K. Sagae, and S. Jayaraman. The significance of recall in automatic metrics for mt evaluation. In *Proceedings of the 6th Conference of the Association for Machine Translation in the Americas (AMTA-2004)*, Washington, DC, September 2004.

[9] Gregor Leusch, Nicola Ueffing, and Herman Ney. String-to-string distance measure with applications to machine translation evaluation. In *Proceedings of MT Summit IX*, pages 240–247, New Orleans, LA, September 2003.

[10] M. Marcus, G. Kim, M. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger. The Penn Treebank: Annotating predicate argument structure. In *ARPA Human Language Technology Workshop*, ARPA Human Language Technology Workshop, 1994.

[11] I. Mel'cuk. *Dependency Syntax: Theory and Practice*. The SUNY Press, Albany, NY, 1988.

[12] S. Niessen, F. J. Och, G. Leusch, and H. Ney. An evaluation tool for machine translation: Fast evaluation for machine translation research. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC-2000)*, pages 39–45, Athens, Greece, 2000.

[13] Frank Smadja, Kathleen R. McKeown, and Vasileios Hatzivassiloglou. Translating collocations for bilingual lexicons: A statistical approach. *Computational Linguistics*, 22(1):1–38, 1996.

[14] K.-Y. Su, M.-W. Wu, and J.-S. Chang. A new quantitative quality measure for machine translation systems. In *Proceedings of the fifteenth International Conference on Computational Linguistics (COLING-92)*, pages 433–439, Nantes, France, 1992.