Meteor++: Incorporating Copy Knowledge into Machine Translation Evaluation

Yinuo Guo, Chong Ruan, Junfeng Hu*
Key Laboratory of Computational Linguistics, School of EECS, Peking University
{gyn0806, pkurc, hujf}@pku.edu.cn

Abstract

In machine translation evaluation, a good candidate translation can be regarded as a paraphrase of the reference. We notice that some words are always copied during paraphrasing, which we call **copy knowledge**. Considering the stability of such knowledge, a good candidate translation should contain all these words appeared in the reference sentence. Therefore, in this participation of the WMT'2018 metrics shared task we introduce a simple statistical method for copy knowledge extraction, and incorporate it into Meteor metric, resulting in a new machine translation metric **Meteor++**. Our experiments show that Meteor++ can nicely integrate copy knowledge and improve the performance significantly on WMT17 and WMT15 evaluation sets.

1 Introduction

Automatic Metrics for machine translation (MT) evaluation have received significant attention in the past few years. MT evaluation measures how close machine-generated translations are to professional human translations, which can be treated as paraphrase evaluation except when the candidates are identical to references. The main difference is that MT evaluation only takes the correctness into consideration while paraphrase evaluation also focuses on diversity.

According to some previous studies on paraphrasing, we find that paraphrasing knowledge can be divided into two categories: copy knowledge and paraphrasable knowledge. The former reflects stable information which tends to keep intact during paraphrasing, while the latter can be paraphrased in various ways. There are some previous researches taking account of copy mechanism (Vinyals et al., 2015; Gu et al., 2016; See et al., 2017; Li et al., 2017) in text generation. And in this paper, we extend the idea of copy from generation to MT evaluation.

Firstly, we give an introduction to copy knowledge extraction on paraphrase corpus, and then propose Meteor++ incorporated with it based on Meteor. Our experiment results show that Meteor++ has higher Pearson Correlation with human score than Meteor on WMT evaluation sets and demonstrate the efficacy of copy knowledge.

2 Background

Various metrics for MT evaluation have been proposed and the widely used metrics are BLEU (Papineni et al., 2002) and Meteor (Banerjee and Lavie, 2005; Denkowski and Lavie, 2011, 2014). The main principle behind BLEU is the measurement of n-gram overlapping between the words produced by the machine and the human translation references at the corpus level. BLEU emphasizes precision and not take recall into account directly while Meteor not only combines the two but also gives a higher weight to recall in general. We choose Meteor in this paper because recall is extremely important for assessing the quality of MT output, as it reflects to what degree the translation covers the entire content of the source sentence.

The Meteor metric has been shown to have high correlation with human judgments in evaluation such as the 2010 ACL Workshop on Statistical Machine Translation and NIST Metrics MATR (Callison-Burch et al., 2010). It is based on general concept of flexible unigram matching, unigram precision and unigram recall, including the match of words that are simple morphological variants of each other by the identical stem and words that are synonyms of each other. Meteor firstly conduct an alignment include several stages (exact, stem, synonym and paraphrase) with different weight between two sentences. Then cal-

Proceedings of the Third Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 753–758

https://doi.org/10.18653/v1/W18-6455

word	c / p	word	c / p	word	c / p	word	c / p
instagram	877/.950	meth	378/.923	dandruff	20/1.0	communism	21/1.0
gmail	725/.905	python	393/.908	edmonton	104/1.0	algebra	24/1.0
traffic	628/.936	shotguns	549/.961	auckland	104/1.0	airprint	97/1.0
youtube	621/.944	linux	173/.913	vinegar	31/1.0	chess	62/1.0
java	476/.901	earthquake	277/.981	cellulite	29/1.0	officejet	97/1.0
kerala	352/.989	hacker	267/.902	hamsters	75/1.0	hamsters	75/1.0
macbook	333/.931	kvpy	258/1.0	bermuda	63/1.0	monday	24/1.0
sahara	306/.935	yahoo	207/.913	salman	23/1.0	forex	36/1.0

Table 1: Quora "copy-words" examples, **c** means raw count and **p** means co-occurrence probability, totally we extract 427 " copy-words " with 20 as the **c** threshold and 0.85 as the **p** threshold. Note that all the words are in their lower cases.

word	c / p	word	c / p	word	c / p	word	c / p
president	37/.833	10	27/.815	2016	13/1.0	hamas	4/1.0
police	36/.889	women	23/.913	hepatitis	3/1.0	romania	4/1.0
world	35/.886	economy	22/.867	john	3/1.0	washington	3/1.0
russia	34/.824	government	22/.818	kingfisher	5/1.0	hundreds	7/1.0
million	32/.813	clinton	22/.910	garland	9/1.0	victim	3/1.0
trump	31/.968	thursday	20/1.0	local	14/1.0	facebook	11/1.0
putin	18/1.0	week	17/.941	ukraine	9/1.0	french	7/1.0

Table 2: WMT "copy-words" examples, **c** means raw count and **p** means co-occurrence probability, we select the candidates with the human scores greater or equal to 0.7 and combine them with their references as paraphrase pairs. Finally, we filter out 1088 paraphrase pairs with a vocabulary of 4619 words. Totally we extract 268 "copy-words" with 2 as **c** threshold and 0.8 as the **p** threshold. Note that all the words are in their lower cases.

culate weighted precision P and recall R. For each matcher (m_i), it counts the number of content and function words covered by matches of ith type in the candidate ($m_i(h_c)$, $m_i(h_f)$) and reference ($m_i(r_c)$, $m_i(r_f)$), $|h_f|$ and $|r_f|$ mean the total number of function words in candidate and reference, $|h_c|$ and $|r_c|$ mean the total number of content words in candidate and reference.

$$P = \frac{\sum_i w_i \cdot (\delta \cdot m_i(h_c) + (1-\delta) \cdot m_i(h_f))}{\delta \cdot |h_c| + (1-\delta) \cdot |h_f|} \tag{1}$$

$$R = \frac{\sum_i w_i \cdot (\delta \cdot m_i(r_c) + (1-\delta) \cdot m_i(r_f))}{\delta \cdot |r_c| + (1-\delta) \cdot |r_f|} \tag{2}$$

The parameterized harmonic mean of precision P and recall R then calculated:

$$F_{mean} = \frac{P \cdot R}{\alpha \cdot P + (1-\alpha) \cdot R} \tag{3}$$

To account for gaps and differences in word order, a fragmentation penalty is calculated using the total number of matched words (m, averaged over hypothesis and reference) and number of chunks(ch):

$$Pen = \gamma \cdot (\frac{ch}{m})^{\beta} \tag{4}$$

The Meteor score is then calculated:

$$Score = (1 - Pen) \cdot F_{mean} \tag{5}$$

The parameters α, β, γ, δ and $w_i ... w_n$ are tuned to maximize correlation with human judgments.

3 Proposed Method

3.1 Copy Knowledge Extraction

According to our observation of paraphrasing corpus, we discover copy knowledge in which the words always have a high possibility of co-occurrence in paraphrase pairs. In this section, we will introduce a simple statistical method of copy knowledge extraction and present a word list denoted as **"copy-words"**. From this it can be concluded that if there is a missing "copy-word" in the candidate, it discards some important information; on the other hand, if the candidate contains any other extra "copy-words", the two sentences

categories		examples	c / p
Named Entity	LOC	Sahara, Edmonton, Auckland, Russia, Romania, Washington	62/8.9%
	ORG	WTO, OLA, PTE, MIT, HAI	23/3.3%
	PER	Bob, Trump, Salman, Putin, John, Clinton	123/17.7%
	MISC	Instagram, Gmail, communism, algebra, IQ, Monday, French, hundreds, million, 10, Linux, Python, Macbook, Yahoo, XBOX	253/36.4%
OTHERS		traffic, hacker, government, victim, economy	234/33.7%

Table 3: Copy knowledge classification, we combine the copy knowledge of Quora and WMT, and get 695 "copy-words" totally, **c** is the raw count and **p** is the proportion of each type.

are not semantically equivalent. Therefore the recall and precision of copy knowledge play a key role in the quality of translations.

In light of this, we propose a method for copy knowledge extraction in formula (6), p_w means the co-occurrence probability, $C(w)$ means the raw appearance count of word and $C(co_w)$ means co-occurrence count. We select the words whose raw counts and co-occurrence probabilities in high-quality candidates and references exceed certain thresholds (F, P) as "copy-words".

$$\text{"}copy\_words\text{"} = \{w \mid C(w) \geq F \ \wedge \ p_w \geq P\} \tag{6}$$

where

$$p_w = \frac{C(co_w)}{C(w)} \tag{7}$$

Here we test the method described above on the Quora[1] and the WMT datasets. The Quora dataset consists of over $400,000$ lines of potential question duplicate pairs. Each question pair has a binary value that indicates whether the line truly contains a duplicate pair. Here we only use the duplicate question pairs, including $142,963$ paraphrase pairs and a vocabulary of $32,582$ words. The WMT dataset consists of WMT15-17 (Bojar et al., 2017, 2016; Stanojević et al., 2015). We select the candidates with high human scores and combine them with their references as paraphrase pairs. There are 9287 pairs with human scores and only about one thousand pairs are useful. We regard the pairs which have human scores exceed the threshold as useful pairs (here we set the threshold as 0.8). Since the amount of available texts with high human score is quite small, it is still not possible to conclude which words belong to copy knowledge.

[1]https://www.kaggle.com/quora/question-pairs-dataset

Table 1 and Table 2 show part of the copy knowledge extraction results of the Quora and the WMT.

In Table 3, we divide the copy knowledge into several categories, and find that it is mainly composed of locations, persons, organizations, miscellaneousness and some others. We label these 695 ($427 + 268$) "copy-words" manually and see that about 67% of them are named entities. In general, named entity occupies a large proportion.

3.2 Model

Inspired by the observation of copy knowledge, we propose Meteor++ based on Meteor. In Meteor++, we incorporate copy knowledge into precision P and recall R indirectly. Specifically, we give penalties to the following two conditions from the perspective of recall and precision:

- **Recall :** there exist some "copy-words" only in references but not in candidates.
- **Precision :** there exist some "copy-words" only in candidates but not in references.

The candidates suffer the first condition may discard some important information, and the second may add some other extra information. We propose to correct the formulation of precision P and recall R in Meteor as following:

$$\tilde{P} = P \cdot \frac{X + \sum_i m_i(h_p)}{X + |h_p|} \tag{8}$$

$$\tilde{R} = R \cdot \frac{X + \sum_i m_i(r_p)}{X + |r_p|} \tag{9}$$

In formula (8), for each matcher (m_i) , which counts the number of "copy-word" covered by matches of i-th type in the candidate ($m_i(h_p)$) and

lang-pair	de-en	fi-en	ru-en	ro-en	cs-en	tr-en	lv-en	zh-en
WMT17	2.102	1.776	2.251	-	1.892	2.201	2.232	2.772
WMT16	1.833	1.988	2.065	2.148	1.499	2.357	-	-
WMT15	1.621	1.816	1.876	-	1.492	-	-	-

Table 4: NE density of each language pair on WMT15-17, NE density means the average count of NE per sentence on each language pair.

	lang-pair	de-en	fi-en	ru-en	ro-en	cs-en	tr-en	lv-en	zh-en	avg
WMT2017	Meteor	.535	.719	.618	-	.550	**.628**	.550	.638	.589
(X = 14)	Meteor++	**.538**	**.720**	**.627**	-	**.552**	.626	**.563**	**.646**	**.593**
WMT2015	Meteor	.612	.628	**.622**	-	.582	-	-	-	.600
(X = 6)	Meteor++	**.626**	**.649**	**.622**	-	**.591**	-	-	-	**.609**

Table 5: Segment-level Pearson correlation of Meteor and Meteor++ for to-English pairs on WMT15 and WMT17, where avg denotes the average Pearson correlation of all language pairs. The parameter X in Meteor++ sets 14 on WMT17 and 8 on WMT15, other parameters are consist of the Meteor Universal.

the reference ($m_i(r_p)$), $|h_p|$ and $|r_p|$ respectively mean the total number of "copy-words" in the candidate and the reference. X is a hyper-parameter used to smooth the results as following:

For Smoothing : In formula (1) and (2), we have already punished the unmatched words, here we only give an appropriate extra penalty to the "copy-words" missing.

Compensation For The Gap : In section 3.1, we only propose a simple statistical method to extract copy knowledge and it still has a long distance from the real copy knowledge.

Likewise, we have the modified recall formula as (9). After that correction, the $\tilde{P}$ and $\tilde{R}$ will substitute for the original P and R in the following calculation.

This two formulas can be regarded as using the precision and the recall of the "copy-words" to punish the entire sentence. If the "copy-words" are not identical in the candidate-reference pair, P and R will be discounted by the formula (8) and (9). We need to obtain a sufficiently high recall and precision of "copy-word" to guarantee the quality of the candidates since the copy knowledge is of greater importance.

4 Experiment Results

4.1 Settings

We evaluate our model on WMT15 and WMT17 metric task evaluation sets by calculating the correlation with the real human scores. The official human judgments of translation quality are collected using direct assessment(DA) (Graham et al., 2013). The direct assessment evaluation protocol give the annotators the reference and one MT output only and ask them to evaluate the translation adequacy of the MT output on an absolute scale.

The WMT datasets totally have 9287 pairs with human scores and after filtering out the lower human score pairs, only about one thousand pairs can be regarded as the paraphrase pairs. As we described in section 3.1, named entity is an important part of copy knowledge and accounts for 67%, here we take named entity as the copy knowledge because of the absence of reference-candidate pairs with high human scores on WMT datasets. And we use NLTK (Loper and Bird, 2002; Bird and Loper, 2004) toolkit to recognize named entities as our "copy-words" in experiments.

Table 4 shows the NE density of each language pair on WMT15-17 datasets and we select the WMT16 evaluation sets as our development sets. Our development experiments show that the parameter X has positive correlation with the NE density. We can see that WMT17 evaluation sets have higher NE density and WMT15 evaluation sets have lower NE density. In the experiments of Table 5, we set $X = 14$ on WMT17 and $X = 8$ on WMT15.

4.2 Results

Table 5 shows the Pearson correlation with the WMT15 and WMT17 direct assessment of translation adequacy at segment-level. We can see that Meteor++ has higher average segment-level Pearson correlation with DA human scores than Meteor on all WMT datasets.

5 Conclusion

In this paper, we describe the submissions of our metric Meteor++ for WMT18 Metrics task in detail. According to the observation of paraphrasing corpus, we discover copy knowledge in which the words keep intact after paraphrasing. We propose a simple statistical method to extract copy knowledge based on the given parallel monolingual paraphrases. Then, we present Meteor++ to examine the method of integrating copy knowledge into MT evaluation based on Meteor. Because words in copy knowledge always have a high possibility to be found in both candidates and references in machine translation, the Meteor++ could process better than Meteor. The experiment results on WMT datasets for each language pair show that Meteor++ has higher average segment-level Pearson correlation with DA human scores than Meteor and demonstrate the efficacy of copy knowledge.

6 Future Work

In this paper, we give a simple statistical method to extract copy knowledge, and propose the Meteor++ incorporate with it. Although it has already demonstrated great promise, we are still in the process of enhancing the metric in the following directions:

Copy Knowledge Extraction: We only propose a simple statistical method to extract copy knowledge which select the words with a high co-occurrence probability in paraphrase pairs. Here we just use bag-of-words to represent sentences and regard the intersection of them as co-occurrence. Therefore the copy knowledge we extract has a long way to go compared to the real copy knowledge. Furthermore, we are considering about constructing an alignment on the large-scale parallel monolingual corpus and then extracting universal copy knowledge based on it for broad use.

Training the hyper-parameter X on Data: The hyper-parameter X was designed to smooth the results and compensate for the gap between the copy knowledge we extract and the real copy knowledge. As our copy knowledge is getting more and more closer to the real copy knowledge, we plan to optimize the formulas by training on a separate data set, and choosing the X formula with the best correlations with human assessment on the training data.

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*, pages 65–72.

Steven Bird and Edward Loper. 2004. Nltk: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 31. Association for Computational Linguistics.

Ondřej Bojar, Yvette Graham, Amir Kamran, and Miloš Stanojević. 2016. Results of the wmt16 metrics shared task. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, volume 2, pages 199–231.

Ondřej Bojar, Yvette Graham, and Amir Kamran. 2017. Results of the WMT17 metrics shared task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Copenhagen, Denmark. Association for Computational Linguistics.

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*, pages 17–53. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the sixth workshop on statistical machine translation*, pages 85–91. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2017. Paraphrase generation with deep reinforcement learning. *arXiv preprint arXiv:1711.00279*.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, pages 63–70. Association for Computational Linguistics.

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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Miloš Stanojević, Amir Kamran, Philipp Koehn, and Ondřej Bojar. 2015. Results of the wmt15 metrics shared task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 256–273.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700.