WMD_O: Fluency-based Word Mover's Distance for Machine Translation Evaluation

Julian Chow, **Pranava Madhyastha** and **Lucia Specia**
Department of Computing
Imperial College London, UK
{julian.chow16,pranava,l.specia}@imperial.ac.uk

Abstract

We propose WMD_O, a metric based on distance between distributions in the semantic vector space. Matching in the semantic space has been investigated for translation evaluation, but the constraints of a translation's word order have not been fully explored. Building on the Word Mover's Distance metric and various word embeddings, we introduce a fragmentation penalty to account for fluency of a translation. This word order extension is shown to perform better than standard WMD, with promising results against other types of metrics.

1 Introduction

Current metrics to automatically evaluate machine translations, such as the popular BLEU (Papineni et al., 2002), are heavily based on string matching. They claim to account for adequacy by checking for overlapping words between the machine translation output and reference translation, and fluency by rewarding matches in sequences of more than one word. This way of viewing adequacy is very limiting; comparing strings makes it harder to evaluate any deviation from the semantics of the original text in the reference or machine translation.

Meteor (Banerjee and Lavie, 2005) relaxes this constraint by allowing matching of lemmas, synonyms or paraphrases. However, this requires linguistic resources to lemmatise the data or lexical databases to fetch synonyms/paraphrases, which do not exist for most languages.

Character-based metrics like chrF (Popovic, 2015) and CharacTER (Wang et al., 2016) also relax the exact word match constraint by allowing the matching of characters. However, they ultimately still assume a surface-level similarity between reference and machine translation output.

Chen and Guo (2015) presented a number of experiments where both translation and reference sentences are compared in the embedding space rather than at surface level. They however simply extract these two embedding representations and measure the (cosine) similarity between them, which may account for some overall semantic similarity, but ignores other aspects of translation quality.

A version of Meteor has been proposed that also performs matches at the word embedding space (Servan et al., 2016). Two words are considered to match if their cosine distance in the embedding space is above a certain threshold. In other words, the embeddings are only used to provide this binary decision, rather than to measure overall semantic distance between two sentences. In a similar vein, bleu2vec and ngram2vec (Tttar and Fishel, 2017) are a direct modification of BLEU where fuzzy matches are added to strict matches. The fuzzy match score is implemented via token and n-gram embedding similarities. As we show in Section 4, these metrics do not perform well.

MEANT 2.0 (Lo, 2017) also relies on matching of words in the embedding space, but this is only used to score the similarity between pairs of words that have already been aligned based on their semantic roles, rather than to find the alignments between words.

We suggest a more general way of using distributional representations of words, where distance in the semantic space is viewed as a global decision between the entire machine and reference translations. More specifically, we propose an adaptation of a powerful and flexible metric that operates on the semantic space: Word Mover's Distance (WMD) (Kusner et al., 2015). WMD is an instance of the Earth Mover's Distance transportation problem that calculates the most efficient way to transform one distribution onto another.

Proceedings of the Fourth Conference on Machine Translation (WMT), Volume 2: Shared Task Papers (Day 1) pages 494–500
Florence, Italy, August 1-2, 2019.

Adjustments to EMD have been used previously to create evaluation metrics based on word embeddings and word positions (Echizen'ya et al., 2019). Likewise, using vector word embeddings as an indicator of similarity and the word embeddings of each text as a distribution, WMD gives the optimal method of transforming the words of one document to the words of another document. WMD does not take word order into account and rather focuses on semantic similarity of word meanings.

WMD has been recently used for the evaluation of image captioning models (Kilickaya et al., 2017; Madhyastha et al., 2019). It proved promising for image captioning evaluation, where word order is less relevant. The same image can be described similarly using different word orders as it is constrained by the image itself. We note that in machine translation evaluation, word order is more important, since the order is constrained by that of the source sentence.

In this paper, we propose WMD_O – an extension to WMD that incorporates word order. We show that this metric outperforms the standard WMD and performs on par or better than most state-of-the-art evaluation metrics.

2 Method

In this section we describe the original WMD distance metric and its extension to account for word order.

2.1 WMD

Word Mover's Distance (WMD) (Kusner et al., 2015) makes use of vectorial relationships between word embeddings to compute distance between two text documents. In essence, WMD captures the minimal distance required to move words from the first document to words in the second document.

Let $X \in \mathcal{R}^{n \times d}$ be a d-dimensional word embedding matrix for a vocabulary of n words. Let $x_i \in \mathcal{R}^d$ be a d-dimensional representation of i^{th} word. Assume two documents A and B with d^a and d^b as the normalized bag-of-words (BOW) vectors, k-dimensional vectors for the respective documents, where ${d_j}^a$ is the number of times word j occurs in A (normalized by all words appearing in A). Note that stop words are removed from documents; only content words are retained.

Kusner et al. (2015) propose the word travel cost, that is the cost of moving words from T_i^a to T_j^b, as the measure of word dissimilarity, using the Euclidean distance between the embeddings corresponding to words. More precisely, the cost associated is defined as:

$$c(i, j) = \|x_i - x_j\|_2^2 \,, \tag{1}$$

This allows documents with many closely related words to have smaller distances than documents with very dissimilar words. WMD defines a transport matrix $T \in \mathcal{R}^{n \times n}$,

where T_{ij} contains information about the proportion of d_i^a that needs to be transported to d_j^b. Formally, WMD computes T that optimizes:

$$D(d^a, d^b) = \min_{T \geq 0} \sum_{i,j=1}^{n} T_{ij} c(i, j), \tag{2}$$

such that: $\sum_{j=1}^{n} T_{ij} = d_i^a$ and $\sum_{i=1}^{n} T_{ij} = d_j^b$, $\forall\, i, j$. Here, the normalized bag-of-words distribution of the documents d^a and d^b contains a combined vocabulary from d^a and d^b resulting in a square transport matrix T of dimensionality $n \times n$.

We note that Kusner et al. (2015) remove stop words and retain only content words before computing WMD, as stop words are generally less relevant for capturing content specific similarity between documents. In our implementation, we include the stop words in order to capture a more coherent distance.

2.2 WMD with word order

Evaluation of translation candidates generally takes into account fluency as well as adequacy to form a judgment. As described in previous section, the standard WMD does not take word order into account. We introduce a modified version which includes a specialized penalty that is intended to penalize for words occurring in a different order from the reference translation. This modification adds a notion of fluency on top of the original WMD metric, which is crucial in matching the multifaceted approach of human translation evaluation.

The word order penalty is applied after calculation of the standard WMD score. Our proposal for penalty is similar to the notion of fragmentation penalty of Meteor (Banerjee and Lavie, 2005), which separates word matches into chunks in order to prevent the metric from doubly-penalising a translation for having out of order consecutive words. These chunks are defined as a group of unigrams which are adjacent in both reference and

machine translation. The longer each chain of n-grams is, the fewer the chunks, so if the entire machine translation matches the reference in consecutive order there is only one chunk. Figure 1 is an illustration of the use of chunks. The matched unigrams for "the president" and "spoke loudly" are in the same order in both sentences, giving two chunks for this translation, fragmented by the word "then".

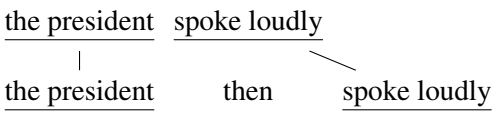

Figure 1: An example of chunks.

This type of word order penalty is necessary to deal with examples such as that of Figure 2. The sentence gets a perfect WMD score because all of its words align exactly to another one in the vector space, with no regard to its fluency. With a fragmentation penalty, this type of situation would see the score get worse because of its different sentence structure to the reference.

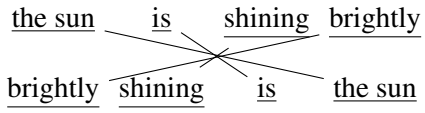

Figure 2: The WMD score for this sentence pair is 0.0.

The penalty is formulated as:

$$\text{Penalty} = \frac{c}{u_m} \tag{3}$$

where c is the number of chunks and u_m is the number of unigrams in the machine translation.

This penalty is weighted by a value δ. and is formulated as:

$$\text{Weight} = \delta \times \text{Penalty} \tag{4}$$

We also observed that, in many cases, the simple penalty in Equation 4 can further be augmented with a modification that rewards sentences which are largely contiguous. We modify Equation 4 such that sentences with fewer chunks are rewarded and sentences with more chunks are penalized. We empirically found that $\frac{1}{2}$ is optimal for such a realization. With this modification, our fluency based word mover's distance (WMD_O) is defined as:

$$\text{WMD}_\text{O} = \text{WMD} - \delta(\frac{1}{2} - \text{Penalty}) \tag{5}$$

We also observe that, in most cases, the optimal weight seems to be 0.2.

3 Experimental settings

We performed experiments to verify the performance of the proposed metric, comparing the metric's results against human annotations to measure a level of correlation. We used the PyEMD wrapper (Mayner, 2019) for calculating the WMD, based on (Pele and Werman, 2008, 2009). We did not remove any stopwords as these are important to fluency. We also use Cosine rather than Euclidean distance to calculate distance between word embeddings as magnitude of the vectors is not as important in such high dimensions.

3.1 Datasets

We used the WMT17 segment-level into-English datasets for our experiments (Bojar et al., 2017). This has data from seven different source languages, with 560 different texts each. Every text carries a reference translation and a machine translation, with a human annotation labelling how closely the machine translation relates to the reference.

3.2 Word embeddings

Many pre-trained word embeddings are available for English. Since word2vec embeddings have been shown to work well with WMD, this was our starting point as the embeddings used to develop the metric. We used a freely-available pre-trained model of 300 dimensions trained on approximately 100 billion words from news articles (Mikolov et al., 2013). This model had a vocabulary size of 3 million. While large, there were still many instances of out-of-vocabulary (OOV) words in the WMT17 dataset alone. Some of this can be attributed to incomplete translations; many of the missing words were foreign words in the source language. Other instances were proper nouns which had not been seen in the pre-trained embeddings vocabulary, as well as numerical values for the same reason.

To tackle OOV, we tried several different approaches. One was to assign a single random vector as an OOV vector, using the same vector for

every instance of a missing word. For these experiments, we used the vector of all 0s, as this seemed the most neutral. Another was to have a random vector for each OOV word and store it in a dictionary, calling on the same value whenever the OOV word is encountered again. In the same vein, one setting was to generate this vector by taking an average of five random vectors in the embedding.

An alternative approach we also pursued was to use a different set of embeddings. FastText (Mikolov et al., 2018) is a type of embedding which is able to produce embeddings for words not part of the vocabulary. This utilises vectors from of substrings of characters contained in the missing word, adding them together so even vectors for misspelled words or a concatenation of words can be produced. Again, a pre-trained model, also of 300 dimensions and trained on news articles was used here. We also fine-tuned this model to produce another set of embeddings, using monolingual training data from the WMT19 news translation task. The experiments with these embeddings were done with and without the FastText character n-gram method of solving OOVs.

All of these approaches were used to test the metric against human scores, the results of which can be seen in Section 4.

4 Results

The results of these experiments are shown in Tables 1 to 4. Each row in a table corresponds to an experimental setting, while each column represents one of the seven language pairs. The value of each cell represents the Pearson correlation with of the metric's score with the given human score, with a higher value suggesting better agreement with the gold standard human evaluation.

Table 1 shows the results of the different OOV strategies, all using the pre-trained word2vec embedding and the standard WMD metric. Out of these strategies, the same random vector for all OOVs came out top by a small margin.

Table 2 looks at the effect of using different embeddings on results and OOV rate, including with and without the n-gram method of FastText to resolve OOVs. We can see that the pre-trained FastText vectors with the OOV resolution strategy of the same vector for all OOV had the best performance, but only marginally over a random vector for each OOV. A different vector choice might be better for different embeddings, but for the purposes of further experiments with this dataset the zero vector was used. It also shows that the FastText embeddings perform better than the word2vec embedding with the same OOV resolution strategy, suggesting a difference in quality of vectors.

Table 3 presents the experimental results of WMD and the WMD word order metrics for different values of δ. These experiments used the pre-trained FastText vectors with a zero vector for all OOV. It shows that the WMD word order metric performs better than the standard WMD metric in the majority of language pairs.

Combining these results, we find that the best performing iteration of our metric for all language pairs is the word order version of WMD, with δ at 0.2. This is using the pre-trained FastText embedding, with the zero vector used for each OOV word. However, it should be noted that some language pairs perform slightly better with a higher or lower δ; this is reflected in the next table with the "ideal" parameter.

We compare this to the rest of the results from the WMT17 metrics task in Table 4; it shows that our metric performs at a similar level or better than most evaluation metrics. Of the metrics which do better than WMD_O. Blend and AutoDA are trained metrics, which are not the most practical when applied to larger datasets as they rely on human annotated training data. MEANT is a metric that does very well for most language combinations. It also uses word embeddings to score matching words, but it is not clear whether the benefit comes from this or from other components in the metric. Overall, this metric has a very large number of steps that rely on linguistic resources, and its code is not available.

5 Analysis

We plot two examples of the distributions of human and WMD_O metric scores in Figures 3 and 4. The results for Finnish-English were fairly strong, but those for Latvian-English had a few more anomalies.

The metric performs sufficiently with reference and machine translated outputs which were largely of a similar length, as the influence of each word was not overbearing on the metric's end result. This can be seen in the results for Finnish to English, which are quite consistent.

Our metric struggled more with bad translations

	cs-en	de-en	fi-en	lv-en	ru-en	tr-en	zh-en
Same vector for all OOV	0.513	0.531	0.689	0.505	0.562	0.561	0.595
Random vector per OOV	0.513	0.531	0.687	0.501	0.560	0.557	0.591
Average of 5 random vectors	0.500	0.534	0.678	0.492	0.563	0.557	0.572

Table 1: Performance of OOV strategies with standard WMD and word2vec.

	cs-en	de-en	fi-en	lv-en	ru-en	tr-en	zh-en	OOV (%)
Word2vec (same vector for all OOV)	0.513	0.531	0.689	0.505	0.562	0.561	0.595	0.10
FastText (same vector for all OOV)	0.521	0.536	0.704	0.530	0.571	0.566	0.607	0.22
FastText (random vector per OOV)	0.521	0.536	0.702	0.530	0.571	0.566	0.607	0.22
FastText (n-grams)	0.511	0.542	0.700	0.526	0.572	0.577	0.583	0
FastText finetuned (n-grams)	0.485	0.525	0.671	0.513	0.546	0.538	0.597	0

Table 2: Performance of different embeddings on standard WMD, including OOV rate.

	cs-en	de-en	fi-en	lv-en	ru-en	tr-en	zh-en
WMD	0.521	0.536	0.704	0.530	0.571	0.566	0.607
WMD_O, $\delta = 0.05$	0.528	0.544	0.709	0.537	0.580	0.585	0.616
WMD_O, $\delta = 0.1$	**0.531**	**0.546**	**0.710**	0.541	**0.585**	0.600	0.621
WMD_O, $\delta = 0.2$	0.530	0.542	0.705	**0.543**	**0.585**	0.620	**0.623**
WMD_O, $\delta = 0.3$	0.525	0.534	0.696	0.540	0.579	0.631	0.621
WMD_O, $\delta = 0.4$	0.518	0.525	0.686	0.535	0.572	**0.637**	0.616

Table 3: Performance of different WMD implementations with pre-trained FastText and same vector strategy. Bolded value signify the best performing metric for each language pair.

	cs-en	de-en	fi-en	lv-en	ru-en	tr-en	zh-en
AUTODA	0.499	0.543	0.673	0.533	0.584	0.625	0.583
BEER	0.511	0.530	0.681	0.515	0.577	0.600	0.582
BLEND	0.594	0.571	0.733	0.577	0.622	0.671	0.661
BLEU2VEC_SEP	0.439	0.429	0.590	0.386	0.489	0.529	0.526
CHRF	0.514	0.531	0.671	0.525	0.599	0.607	0.591
CHRF++	0.523	0.534	0.678	0.520	0.588	0.614	0.593
MEANT_2.0	**0.578**	**0.565**	0.687	**0.586**	**0.607**	0.596	**0.639**
MEANT_2.0-NOSRL	0.566	0.564	0.682	0.573	0.591	0.582	0.630
NGRAM2VEC	0.436	0.435	0.582	0.383	0.490	0.538	0.520
SENTBLEU	0.435	0.432	0.571	0.393	0.484	0.538	0.512
TREEAGGREG	0.486	0.526	0.638	0.446	0.555	0.571	0.535
UHH_TSKM	0.507	0.479	0.600	0.394	0.465	0.478	0.477
WMD	0.521	0.536	0.704	0.530	0.571	0.566	0.607
WMD_O, $\delta = 0.2$	0.530	0.542	0.705	0.543	0.585	0.620	0.623
WMD_O, $\delta = \text{IDEAL}$	0.531	0.546	**0.710**	0.543	0.585	**0.637**	0.623

Table 4: Performance of different metrics in the WMT17 shared task against the two proposed metrics. Our metrics are highlighted in blue. Trained/ensemble metrics are highlighted in grey. Bolded values signify the best performing non-trained metric for each language pair.

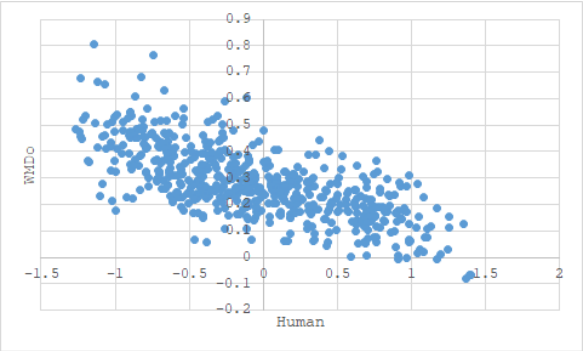

Figure 3: WMD_O against human scores for fi-en

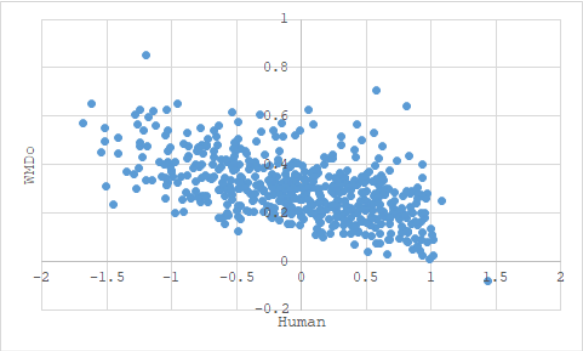

Figure 4: WMD_O against human scores for lv-en

of sentences which were shorter, as each chunk became more pronounced in the penalty, which compounded the bad WMD scores of the nonsensical translation. This was especially evident with poor translations which were comprised largely of retained foreign words. An example of this is from the Latvian to English set; one of the machine translations was "Pann uzkars oil" for the reference "Heat oil in a frying-pan". The penalty could be adjusted in the future to account for sentence length.

6 Conclusions

We have proposed a novel method of evaluating machine translations, focusing on word embeddings and the semantic space. Our metric implementing a word order weighting achieved strong performance in relation to other state-of-the-art metrics and the standard WMD metric. From this we can conclude that semantic spaces are a viable approach to assessing machine translations.

In terms of experimental settings, we found that using the n-gram approach of FastText did not significantly outperform initialising a random vector for each OOV word, although the higher quality FastText embeddings proved to be more accurate than the older word2vec embeddings. These settings, along with the value of δ, may vary for different datasets. This may be because the WMT17 dataset had a large number of foreign words, which would not make much sense to use n-grams to piece back together. In addition, the finetuned FastText embedding might have had suboptimal training parameters, leading to its poorer performance. It can also be seen that different values of δ work better on certain language pairs; this may have to be a value tuned per language pair rather than a catch-all value.

This work within semantic spaces can also be extended to other translation tasks; as comparisons of two segments are performed within the currently monolingual vector space, future translation evaluations could make use of cross-lingual word embeddings, which carry vectors for different languages in the same space. This could potentially allow translation evaluations to be done directly from the source text to the machine translation, without the human evaluation in between by using a vector space combining the source and target language. Work into cross-lingual embeddings has been growing in recent years (Conneau et al., 2017) and this metric could be used to leverage the potential of this area in the future of automatic translation evaluation. We will provide an open source implementation of WMD_O (Chow, 2019).

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pages 65–72.

Ondrej Bojar, Yvette Graham, and Amir Kamran. 2017. Results of the wmt17 metrics shared task. Proceedings of the Second Conference on Machine Translation, page 489513.

Boxing Chen and Hongyu Guo. 2015. Representation based translation evaluation metrics. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 150–155, Beijing, China. Association for Computational Linguistics.

Julian Chow. 2019. Wmdo. https://github.com/julianchow/WMDO.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017.

Word translation without parallel data. arXiv preprint arXiv:1710.04087.

Hiroshi Echizen'ya, Kenji Araki, and Eduard Hovy. 2019. Word embedding-based automatic mt evaluation metric using word position information. Proceedings of NAACL-HLT, page 18741883.

Mert Kilickaya, Aykut Erdem, Nazli Ikizler-Cinbis, and Erkut Erdem. 2017. Re-evaluating automatic metrics for image captioning. Proceedings of EACL 2017, pages 199–209.

Matt J Kusner, Yu Sun, Nicholas I Kolkin, and Kilian Q Weinberger. 2015. From word embeddings to document distances. Proceedings of the 32nd International Conference on International Conference on Machine Learning, 37:957–966.

Chi-kiu Lo. 2017. Meant 2.0: Accurate semantic mt evaluation for any output language. In Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers, pages 589–597, Copenhagen, Denmark. Association for Computational Linguistics.

Pranava Madhyastha, Josiah Wang, and Lucia Specia. 2019. VIFIDEL: evaluating the visual fidelity of image descriptions. In Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL).

William Mayner. 2019. Fast emd for python: a wrapper for pele and werman's c++ implementation of the earth mover's distance metric. https://github.com/wmayner/pyemd.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. ICLR Workshop.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018).

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 311–318.

Ofir Pele and Michael Werman. 2008. A linear time histogram metric for improved sift matching. In Computer Vision–ECCV 2008, pages 495–508. Springer.

Ofir Pele and Michael Werman. 2009. Fast and robust earth mover's distances. In 2009 IEEE 12th International Conference on Computer Vision, pages 460–467. IEEE.

Maja Popovic. 2015. CHRF: character n-gram F-score for automatic MT evaluation. Proceedings of the Tenth Workshop on Statistical Machine Translation, pages 392–395.

Christophe Servan, Alexandre Berard, Zied Elloumi, Hervé Blanchon, and Laurent Besacier. 2016. Word2Vec vs DBnary: Augmenting METEOR using Vector Representations or Lexical Resources? Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 1159–1168.

Andre Tttar and Mark Fishel. 2017. bleu2vec: the painfully familiar metric on continuous vector space steroids. Proceedings of the Second Conference on Machine Translation, page 619622.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. 2016. CharacTER: Translation Edit Rate on Character Level. Proceedings of the First Conference on Machine Translation.