Pairwise Neural Machine Translation Evaluation

Francisco Guzmán **Shafiq Joty** **Lluís Màrquez** and **Preslav Nakov**
ALT Research Group
Qatar Computing Research Institute — HBKU, Qatar Foundation
{fguzman,sjoty,lmarquez,pnakov}@qf.org.qa

Abstract

We present a novel framework for machine translation evaluation using neural networks in a pairwise setting, where the goal is to select the better translation from a pair of hypotheses, given the reference translation. In this framework, lexical, syntactic and semantic information from the reference and the two hypotheses is compacted into relatively small distributed vector representations, and fed into a multi-layer neural network that models the interaction between each of the hypotheses and the reference, as well as between the two hypotheses. These compact representations are in turn based on word and sentence embeddings, which are learned using neural networks. The framework is flexible, allows for efficient learning and classification, and yields correlation with humans that rivals the state of the art.

1 Introduction

Automatic machine translation (MT) evaluation is a necessary step when developing or comparing MT systems. *Reference*-based MT evaluation, i.e., comparing the system output to one or more human reference translations, is the most common approach. Existing MT evaluation measures typically output an absolute quality score by computing the similarity between the machine and the human translations. In the simplest case, the similarity is computed by counting word n-gram matches between the translation and the reference. This is the case of BLEU (Papineni et al., 2002), which has been the standard for MT evaluation for years. Nonetheless, more recent evaluation measures take into account various aspects of linguistic similarity, and achieve better correlation with human judgments.

Having absolute quality scores at the sentence level allows to rank alternative translations for a given source sentence. This is useful, for instance, for statistical machine translation (SMT) parameter tuning, for system comparison, and for assessing the progress during MT system development. The quality of automatic MT evaluation metrics is usually assessed by computing their correlation with human judgments. To that end, quality rankings of alternative translations have been created by human judges. It is known that assigning an absolute score to a translation is a difficult task for humans. Hence, ranking-based evaluations, where judges are asked to rank the output of 2 to 5 systems, have been used in recent years, which has yielded much higher inter-annotator agreement (Callison-Burch et al., 2007).

These human quality judgments can be used to train automatic metrics. This supervised learning can be oriented to predict absolute scores, e.g., using regression (Albrecht and Hwa, 2008), or rankings (Duh, 2008; Song and Cohn, 2011). A particular case of the latter is used to learn in a pairwise setting, i.e., given a reference and two alternative translations (or hypotheses), the task is to decide which one is better. This setting emulates closely how human judges perform evaluation assessments in reality, and can be used to produce rankings for an arbitrarily large number of hypotheses. In this pairwise setting, the challenge is to learn, from a pair of hypotheses, which are the features that help to discriminate the better from the worse translation. Although the pairwise setting does not produce absolute quality scores (i.e., it is not an evaluation metric applicable to a single translation), it is useful and arguably sufficient for most evaluation and MT development scenarios.[1]

[1] We do not argue that the pairwise approach is better than the direct estimation of human quality scores. Both approaches have pros and cons; we see them as complementary.

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 805–814,
Beijing, China, July 26-31, 2015.

Recently, Guzmán et al. (2014a) presented a learning framework for this pairwise setting, based on preference kernels and support vector machines (SVM). They obtained promising results using syntactic and discourse-based structures. However, using convolution kernels over complex structures comes at a high computational cost both at training and at testing time because the use of kernels requires that the SVM operate in the much slower dual space. Thus, some simplification is needed to make it practical. While there are some solutions in the kernel-based learning framework to alleviate the computational burden, in this paper we explore an entirely different direction.

We present a novel neural-based architecture for learning in the pairwise setting for MT evaluation. Lexical, syntactic and semantic information from the reference and the two hypotheses is compacted into relatively small distributed vector representations and fed into the input layer, together with a set of individual real-valued features coming from simple pre-existing MT evaluation metrics. A hidden layer, motivated by our intuitions on the pairwise ranking problem, is used to capture interactions between the relevant input components. Finally, we present a task-oriented cost function, specifically tailored for this problem.

Our evaluation results on the *WMT12 metrics task* benchmark datasets (Callison-Burch et al., 2012) show very high correlation with human judgments. These results clearly surpass (Guzmán et al., 2014a) and are comparable to the best previously reported results for this dataset, achieved by DiscoTK (Joty et al., 2014), which is a much heavier combination-based metric.

Another advantage of the proposed architecture is efficiency. Due to the vector-based compression of the linguistic structure and the relatively reduced size of the network, testing is fast, which would greatly facilitate the practical use of this approach in real MT evaluation and development. Finally, we empirically show that syntactically- and semantically-oriented embeddings can be incorporated to produce sizeable and cumulative gains in performance over a strong combination of pre-existing MT evaluation measures (BLEU, NIST, METEOR, and TER). This is promising evidence towards our longer-term goal of defining a general platform for integrating varied linguistic information and for producing more informed MT evaluation measures.

2 Related Work

Contemporary MT evaluation measures have evolved beyond simple lexical matching, and now take into account various aspects of linguistic structures, including synonymy and paraphrasing (Lavie and Denkowski, 2009), syntax (Giménez and Màrquez, 2007; Popović and Ney, 2007; Liu and Gildea, 2005), semantics (Giménez and Màrquez, 2007; Lo et al., 2012), and even discourse (Comelles et al., 2010; Wong and Kit, 2012; Guzmán et al., 2014b; Joty et al., 2014). The combination of several of these aspects has led to improved results in metric evaluation campaigns, such as the *WMT metrics task* (Bojar et al., 2014).

In this paper, we present a general framework for learning to rank translations in the pairwise setting, using information from several linguistic representations of the translations and references. This work has connections with the ranking-based approaches for learning to reproduce human judgments of MT quality. In particular, our setting is similar to that of Duh (2008), but differs from it both in terms of the feature representation and of the learning framework. For instance, we integrate several layers of linguistic information, while Duh (2008) only used lexical and POS matches as features. Secondly, we use information about both the reference and the two alternative translations simultaneously in a neural-based learning framework capable of modeling complex interactions between the features.

Another related work is that of Kulesza and Shieber (2004), in which lexical and syntactic features, together with other metrics, e.g., BLEU and NIST, are used in an SVM classifier to discriminate good from bad translations. However, their setting is not pairwise comparison, but a classification task to distinguish *human-* from *machine-produced* translations. Moreover, in their work, using syntactic features decreased the correlation with human judgments dramatically (although classification accuracy improved), while in our case the effect is positive.

In our previous work (Guzmán et al., 2014a), we introduced a learning framework for the pairwise setting, based on preference kernels and SVMs. We used lexical, POS, syntactic and discourse-based information in the form of tree-like structures to learn to differentiate better from worse translations.

However, in that work we used convolution kernels, which is computationally expensive and does not scale well to large datasets and complex structures such as graphs and enriched trees. This inefficiency arises both at training and testing time. Thus, here we use neural embeddings and multilayer neural networks, which yields an efficient learning framework that works significantly better on the same datasets (although we are not using exactly the same information for learning).

To the best of our knowledge, the application of structured neural embeddings and a neural network learning architecture for MT evaluation is completely novel. This is despite the growing interest in recent years for deep neural nets (NNs) and word embeddings with application to a myriad of NLP problems. For example, in SMT we have observed an increased use of neural nets for language modeling (Bengio et al., 2003; Mikolov et al., 2010) as well as for improving the translation model (Devlin et al., 2014; Sutskever et al., 2014).

Deep learning has spread beyond language modeling. For example, recursive NNs have been used for syntactic parsing (Socher et al., 2013a) and sentiment analysis (Socher et al., 2013b). The increased use of NNs by the NLP community is in part due to (*i*) the emergence of tools such as word2vec (Mikolov et al., 2013a) and GloVe (Pennington et al., 2014), which have enabled NLP researchers to learn word embeddings, and (*ii*) unified learning frameworks, e.g., (Collobert et al., 2011), which cover a variety of NLP tasks such as part-of-speech tagging, chunking, named entity recognition, and semantic role labeling.

While in this work we make use of widely available pre-computed structured embeddings, the novelty of our work goes beyond the type of information considered as input, and resides on the way it is integrated to a neural network architecture that is inspired by our intuitions about MT evaluation.

3 Neural Ranking Model

Our motivation for using neural networks for MT evaluation is twofold. First, to take advantage of their ability to model complex non-linear relationships efficiently. Second, to have a framework that allows for easy incorporation of rich syntactic and semantic representations captured by word embeddings, which are in turn learned using deep learning.

3.1 Learning Task

Given two translation hypotheses t_1 and t_2 (and a reference translation r), we want to tell which of the two is better.[2] Thus, we have a binary classification task, which is modeled by the class variable y, defined as follows:

$$y = \begin{cases} 1 & \text{if } t_1 \text{ is better than } t_2 \text{ given } r \\ 0 & \text{if } t_1 \text{ is worse than } t_2 \text{ given } r \end{cases} \quad (1)$$

We model this task using a feed-forward neural network (NN) of the form:

$$p(y|t_1, t_2, r) = \text{Ber}(y|f(t_1, t_2, r)) \quad (2)$$

which is a Bernoulli distribution of y with parameter $\sigma = f(t_1, t_2, r)$, defined as follows:

$$f(t_1, t_2, r) = \text{sig}(\mathbf{w_v^T}\phi(t_1, t_2, r) + b_v) \quad (3)$$

where sig is the sigmoid function, $\phi(x)$ defines the transformations of the input x through the hidden layer, $\mathbf{w_v}$ are the weights from the hidden layer to the output layer, and b_v is a bias term.

3.2 Network Architecture

In order to decide which hypothesis is *better* given the tuple (t_1, t_2, r) as input, we first map the hypotheses and the reference to a fixed-length vector $[\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, \mathbf{x}_r]$, using syntactic and semantic embeddings. Then, we feed this vector as input to our neural network, whose architecture is shown in Figure 1.

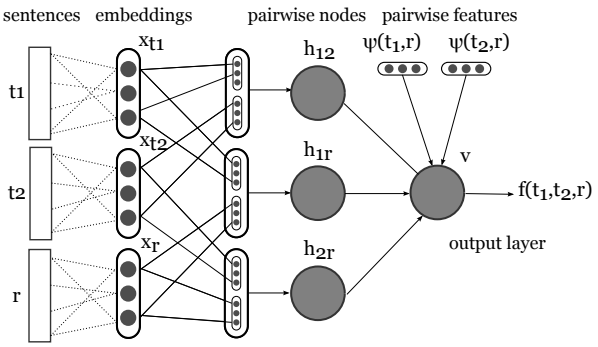

Figure 1: Overall architecture of the neural network.

In our architecture, we model three types of interactions, using different groups of nodes in the hidden layer. We have two *evaluation* groups $\mathbf{h_{1r}}$ and $\mathbf{h_{2r}}$ that model how similar each hypothesis t_i is to the reference r.

[2] In this work, we do not learn to predict ties, and ties are excluded from our training data.

The vector representations of the hypothesis (i.e., $\mathbf{x}_{t1}$ or $\mathbf{x}_{t2}$) together with the reference (i.e., $\mathbf{x}_r$) constitute the input to the hidden nodes in these two groups. The third group of hidden nodes $\mathbf{h_{12}}$, which we call *similarity* group, models how close t_1 and t_2 are. This might be useful as highly similar hypotheses are likely to be comparable in quality, irrespective of whether they are good or bad in absolute terms.

The input to each of these groups is represented by concatenating the vector representations of the two components participating in the interaction, i.e., $\mathbf{x_{1r}} = [\mathbf{x}_{t_1}, \mathbf{x}_r]$, $\mathbf{x_{2r}} = [\mathbf{x}_{t_2}, \mathbf{x}_r]$, $\mathbf{x_{12}} = [\mathbf{x}_{t_1}, \mathbf{x}_{t_2}]$. In summary, the transformation $\phi(t_1, t_2, r) = [\mathbf{h_{12}}, \mathbf{h_{1r}}, \mathbf{h_{2r}}]$ in our NN architecture can be written as follows:

$$\begin{aligned}\mathbf{h_{1r}} &= g(\mathbf{W_{1r}}\mathbf{x_{1r}} + \mathbf{b_{1r}}) \\ \mathbf{h_{2r}} &= g(\mathbf{W_{2r}}\mathbf{x_{2r}} + \mathbf{b_{2r}}) \\ \mathbf{h_{12}} &= g(\mathbf{W_{12}}\mathbf{x_{12}} + \mathbf{b_{12}})\end{aligned}$$

where $g(.)$ is a non-linear activation function (applied component-wise), $\mathbf{W} \in \mathbb{R}^{H \times N}$ are the associated weights between the input layer and the hidden layer, and $\mathbf{b}$ are the corresponding bias terms. In our experiments, we used $\tanh$ as an activation function, rather than sig, to be consistent with how parts of our input vectors were generated.[3]

In addition, our model allows to incorporate external sources of information by enabling *skip arcs* that go directly from the input to the output, skipping the hidden layer. In our setting, these arcs represent pairwise similarity features between the translation hypotheses and the reference (e.g., the BLEU scores of the translations). We denote these pairwise external feature sets as $\psi_{1r} = \psi(t_1, r)$ and $\psi_{2r} = \psi(t_2, r)$. When we include the external features in our architecture, the activation at the output, i.e., eq. (3), can be rewritten as follows:

$$f(t_1, t_2, r) = \mathrm{sig}(\mathbf{w_v^T}[\phi(t_1, t_2, r), \psi_{1r}, \psi_{2r}] + b_v)$$

3.3 Network Training

The negative log likelihood of the training data for the model parameters $\theta = (\mathbf{W_{12}}, \mathbf{W_{1r}}, \mathbf{W_{2r}}, \mathbf{w_v}, \mathbf{b_{12}}, \mathbf{b_{1r}}, \mathbf{b_{2r}}, b_v)$ can be written as follows:

$$J_\theta = -\sum_n y_n \log \hat{y}_{n\theta} + (1 - y_n) \log (1 - \hat{y}_{n\theta}) \quad (4)$$

[3]Many of our input representations consist of word embeddings trained with neural networks that used $\tanh$ as an activation function.

In the above formula, $\hat{y}_{n\theta} = f_n(t_1, t_2, r)$ is the activation at the output layer for the n-th data instance. It is also common to use a regularized cost function by adding a weight decay penalty (e.g., L_2 or L_1 regularization) and to perform maximum aposteriori (MAP) estimation of the parameters. We trained our network with stochastic gradient descent (SGD), mini-batches and adagrad updates (Duchi et al., 2011), using Theano (Bergstra et al., 2010).

4 Experimental Setup

In this section, we describe the different aspects of our general experimental setup (we will discuss some extensions thereof in Section 6), starting with a description of the input representations we use to capture the syntactic and semantic characteristics of the two hypothesis translations and the corresponding reference, as well as the datasets used to evaluate the performance of our model.

4.1 Word Embedding Vectors

Word embeddings play a crucial role in our model, since they allow us to model complex relations between the translations and the reference using syntactic and semantic vector representations.

Syntactic vectors. We generate a syntactic vector for each sentence using the Stanford neural parser (Socher et al., 2013a), which generates a 25-dimensional vector as a by-product of syntactic parsing using a recursive NN. Below we will refer to these vectors as SYNTAX25.

Semantic vectors. We compose a semantic vector for a given sentence using the average of the embedding vectors for the words it contains (Mitchell and Lapata, 2010). We use pre-trained, fixed-length word embedding vectors produced by (*i*) GloVe (Pennington et al., 2014), (*ii*) COMPOSES (Baroni et al., 2014), and (*iii*) word2vec (Mikolov et al., 2013b).

Our primary representation is based on 50-dimensional GloVe vectors, trained on Wikipedia 2014+Gigaword 5 (6B tokens), to which below we will refer as WIKI-GW25.

Furthermore, we experiment with WIKI-GW300, the 300-dimensional GloVe vectors trained on the same data, as well as with the CC-300-42B and CC-300-840B, 300-dimensional GloVe vectors trained on 42B and on 840B tokens from Common Crawl.

We also experiment with the pre-trained, 300-dimensional word2vec embedding vectors, or WORD2VEC300, trained on 100B words from Google News. Finally, we use COMPOSES400, the 400-dimensional COMPOSES vectors trained on 2.8 billion tokens from ukWaC, the English Wikipedia, and the British National Corpus.

4.2 Tuning and Evaluation Datasets

We experiment with datasets of segment-level human rankings of system outputs from the WMT11, WMT12 and WMT13 Metrics shared tasks (Callison-Burch et al., 2011; Callison-Burch et al., 2012; Macháček and Bojar, 2013). We focus on translating into English, for which the WMT11 and WMT12 datasets can be split by source language: Czech (cs), German (de), Spanish (es), and French (fr); WMT13 also has Russian (ru).

4.3 Evaluation Score

We evaluate our metrics in terms of correlation with human judgments measured using Kendall's τ. We report τ for the individual languages as well as macro-averaged across all languages.

Note that there were different versions of τ at WMT over the years. Prior to 2013, WMT used a strict version, which was later relaxed at WMT13 and further revised at WMT14. See (Macháček and Bojar, 2014) for a discussion. Here we use the strict version used at WMT11 and WMT12.

4.4 Experimental Settings

Datasets: We train our neural models on WMT11 and we evaluate them on WMT12. We further use a random subset of 5,000 examples from WMT13 as a validation set to implement early stopping.
Early stopping: We train on WMT11 for up to 10,000 epochs, and we calculate Kendall's τ on the development set after each epoch. We then select the model that achieves the highest τ on the validation set; in case of ties for the best τ, we select the latest epoch that achieved the highest τ.
Network parameters: We train our neural network using SGD with adagrad, an initial learning rate of $\eta = 0.01$, mini-batches of size 30, and L_2 regularization with a decay parameter $\lambda = 1e^{-4}$. We initialize the weights for our matrices by sampling from a uniform distribution following (Bengio and Glorot, 2010). We further set the size of each of our pairwise hidden layers H to four nodes, and we normalize the input data using min-max to map the feature values to the range $[-1, 1]$.

5 Experiments and Results

The main findings of our experiments are shown in Table 1. Section I of Table 1 shows the results for four commonly-used metrics for MT evaluation that compare a translation hypothesis to the reference(s) using primarily lexical information like word and n-gram overlap (even though some allow paraphrases): BLEU, NIST, TER, and METEOR (Papineni et al., 2002; Doddington, 2002; Snover et al., 2006; Denkowski and Lavie, 2011). We will refer to the set of these four metrics as 4METRICS. These metrics are not tuned and achieve Kendall's τ between 18.5 and 23.5.

Section II of Table 1 shows the results for multi-layer neural networks trained on vectors from word embeddings only: SYNTAX25 and WIKI-GW25. These networks achieve modest τ values around 10, which should not be surprising: they use very general vector representations and have no access to word or n-gram overlap or to length information, which are very important features to compute similarity against the reference. However, as will be discussed below, their contribution is complementary to the four previous evaluation metrics and will lead to significant improvements in combination with them.

Section III of Table 1 shows the results for neural networks that combine the four metrics from 4METRICS with SYNTAX25 and WIKI-GW25. We can see that just combining the four metrics in a flat neural net (i.e., no hidden layer), which is equivalent to a logistic regression, yields a τ of 27.06, which is better than the best of the four metrics by 3.5 points absolute, and also better by over 1.5 points absolute than the best metric that participated at the WMT12 metrics task competition (SPEDE07PP with $\tau = 25.4$). Indeed, 4METRICS is a strong mix that involves not only simple lexical overlap but also approximate matching, paraphrases, edit distance, lengths, etc. Yet, adding to 4METRICS the embedding vectors yields sizeable further improvements: +1.5 and +2.0 points absolute when adding SYNTAX25 and WIKI-GW25, respectively. Finally, adding both yields even further improvements close to τ of 30 (+2.64 τ points), showing that lexical semantics and syntactic representations are complementary.

Section IV of Table 1 puts these numbers in perspective: it lists the τ for the top three systems that participated at WMT12, whose scores ranged between 22.9 and 25.4.

	System	Details	Kendall's τ				
			cz	de	es	fr	AVG
I	**4METRICS: commonly-used individual metrics**						
	BLEU	no learning	15.88	18.56	18.57	20.83	18.46
	NIST	no learning	19.66	23.09	20.41	22.21	21.34
	TER	no learning	17.80	25.31	22.86	21.05	21.75
	METEOR	no learning	20.82	26.79	23.81	22.93	23.59
II	**NN using embedding vectors: syntactic & semantic**						
	SYNTAX25	multi-layer NN	8.00	13.03	12.11	7.42	10.14
	WIKI-GW25	multi-layer NN	14.31	11.49	9.24	4.99	10.01
III	**NN using 4METRICS+ embedding vectors**						
	4METRICS	logistic regression	23.46	29.95	27.49	27.36	27.06
	4METRICS+SYNTAX25	multi-layer NN	26.09	30.58	29.30	28.07	28.51
	4METRICS+WIKI-GW25	multi-layer NN	25.67	32.50	29.21	28.92	29.07
	4METRICS+SYNTAX25+WIKI-GW25	multi-layer NN	26.30	33.19	30.38	28.92	**29.70**
IV	**Comparison to previous results on WMT12**						
	DiscoTK (Joty et al., 2014)	Best on the WMT12 dataset	*na*	*na*	*na*	*na*	30.5
	SPEDE07PP	1st at the WMT12 competition	21.2	27.8	26.5	26.0	25.4
	METEOR*	2nd at WMT12 the competition	21.2	27.5	24.9	25.1	24.7
	(Guzmán et al., 2014a)	Preference kernel approach	23.1	25.8	22.6	23.2	23.7
	AMBER	3rd at the WMT12 competition	19.1	24.8	23.1	24.5	22.9

Table 1: Kendall's tau (τ) on the WMT12 dataset for various metrics. Notes: (*i*) the version of METEOR that took part in the WMT12 competition (marked with * in section IV of the table) is different from the one used in our experiments (section I of the table), (*ii*) values marked as *na* were not reported by the authors.

We can see that 4METRICS is much stronger than the winner at WMT12, and thus arguably a baseline hard to improve upon. While our results are slightly behind those of DiscoTK (Joty et al., 2014), we should note that we only combine four metrics, plus the vectors, while DiscoTK combines over 20 metrics, many of which are costly to compute.

On the other hand, we work in a ranking framework, i.e., we are not interested in producing an absolute score, but in making pairwise decisions only. Mapping these pairwise decisions into an absolute score is challenging and in our experiments it leads to a slight drop in τ (results omitted here to save space).

The only other result on WMT12 by authors working with our pairwise framework is our own previous work (Guzmán et al., 2014a), where we used a preference kernel approach to combine syntactic and discourse trees with lexical information; as we can see, our earlier results are 6 absolute points lower than those we achieve here. Moreover, our NN approach offers advantages over SVMs in terms of computational cost.

Based on these results, we can conclude that word embeddings, whether syntactic or semantic, offer generalizations that efficiently complement very strong metric combinations, and thus should be considered when designing future MT evaluation metrics.

6 Discussion

In this section, we explore how different parts of our framework can be modified to improve its performance, or how it can be extended for further generalization. First, we explore variations of the feature sets from the perspective of both the pairwise features and the embeddings. Then, we analyze the role of the network architecture and of the cost function used for learning.

6.1 Fine-Grained Pairwise Features

We have shown that our NN can integrate syntactic and semantic vectors with scores from other metrics. In fact, ours is a more general framework, where one can integrate the *components of a metric* instead of its score, which could yield better learning. Below, we demonstrate this for BLEU.

BLEU has different components: the n-gram precisions, the n-gram matches, the total number of n-grams (n=1,2,3,4), the lengths of the hypotheses and of the reference, the length ratio between them, and BLEU's brevity penalty. We will refer to this decomposed BLEU as BLEUCOMP. Some of these features were previously used in SIMPBLEU (Song and Cohn, 2011).

The results of using the components of BLEUCOMP as features are shown in Table 2. We see that using a single-layer neural network, which is equivalent to logistic regression, outperforms BLEU by more than +1 τ points absolute.

System	Details	Kendall's τ				
		cz	de	es	fr	AVG
BLEU	no learning	15.88	18.56	18.57	20.83	18.46
BLEUCOMP	logistic regression	18.18	21.13	19.79	19.91	19.75
BLEUCOMP+SYNTAX25	multi-layer NN	20.75	25.32	24.85	23.88	23.70
BLEUCOMP+WIKI-GW25	multi-layer NN	22.96	26.63	25.99	24.10	24.92
BLEUCOMP+SYNTAX25+WIKI-GW25	multi-layer NN	22.84	28.92	27.95	24.90	**26.15**
BLEU+SYNTAX25+WIKI-GW25	*multi-layer NN*	*20.03*	*25.95*	*27.07*	*23.16*	*24.05*

Table 2: Kendall's τ on WMT12 for neural networks using BLEUCOMP, a decomposed version of BLEU. For comparison, the last line shows a combination using BLEU instead of BLEUCOMP.

Source	Alone	Comb.
WIKI-GW25	*10.01*	*29.70*
WIKI-GW300	9.66	**29.90**
CC-300-42B	**12.16**	29.68
CC-300-840B	**11.41**	**29.88**
WORD2VEC300	7.72	29.13
COMPOSES400	**12.35**	28.54

Table 3: Average Kendall's τ on WMT12 for semantic vectors trained on different text collections. Shown are results (*i*) when using the semantic vectors alone, and (*ii*) when combining them with 4METRICS and SYNTAX25. The improvements over WIKI-GW25 are marked in bold.

As before, adding SYNTAX25 and WIKI-GW25 improves the results, but now by a more sizable margin: +4 for the former and +5 for the latter. Adding both yields +6.5 improvement over BLEUCOMP, and almost 8 points over BLEU.

We see once again that the syntactic and semantic word embeddings are complementary to the information sources used by metrics such as BLEU, and that our framework can learn from richer pairwise feature sets such as BLEUCOMP.

6.2 Larger Semantic Vectors

One interesting aspect to explore is the effect of the dimensionality of the input embeddings. Here, we studied the impact of using semantic vectors of bigger sizes, trained on different and larger text collections. The results are shown in Table 3. We can see that, compared to the 50-dimensional WIKI-GW25, 300-400 dimensional vectors are generally better by 1-2 τ points absolute when used in isolation; however, when used in combination with 4METRICS+SYNTAX25, they do not offer much gain (up to +0.2), and in some cases, we observe a slight drop in performance. We suspect that the variability across the different collections is due to a domain mismatch. Yet, we defer this question for future work.

Details	Kendall's τ				
	cz	de	es	fr	AVG
single-layer	25.86	32.06	30.03	28.45	29.10
multi-layer	26.30	33.19	30.38	28.92	**29.70**

Table 4: Kendall's tau (τ) on the WMT12 dataset for alternative architectures using 4METRICS+SYNTAX25+WIKI-GW25 as input.

6.3 Deep vs. Flat Neural Network

One interesting question is how much of the learning is due to the rich input representations, and how much happens because of the architecture of the neural network. To answer this, we experimented with two settings: a single-layer neural network, where all input features are fed directly to the output layer (which is logistic regression), and our proposed multi-layer neural network.

The results are shown in Table 4. We can see that switching from our multi-layer architecture to a single-layer one yields an absolute drop of 0.6 τ. This suggests that there is value in using the deeper, pairwise layer architecture.

6.4 Task-Specific Cost Function

Another question is whether the log-likelihood cost function $J(\theta)$ (see Section 3.3) is the most appropriate for our ranking task, provided that it is evaluated using Kendall's τ as defined below:

$$\tau = \frac{concord. - disc. - ties}{concord + disc. + ties} \quad (5)$$

where *concord.*, *disc.* and *ties* are the number of concordant, disconcordant and tied pairs.

Given an input tuple (t_1, t_2, r), the logistic cost function yields larger values of $\sigma = f(t_1, t_2, r)$ if $y = 1$, and smaller if $y = 0$, where $0 \leq \sigma \leq 1$ is the parameter of the Bernoulli distribution. However, it does not model *directly* the probability when the order of the hypotheses in the tuple is reversed, i.e., $\sigma' = f(t_2, t_1, r)$.

Details	Kendall's τ				
	cz	de	es	fr	AVG
Logistic	26.30	33.19	30.38	28.92	29.70
Kendall	27.04	33.60	29.48	28.54	29.53
Log.+Ken.	26.90	33.17	30.40	29.21	**29.92**

Table 5: Kendall's tau (τ) on WMT12 for alternative cost functions using 4METRICS+SYNTAX25+WIKI-GW25.

For our specific task, given an input tuple (t_1, t_2, r), we want to make sure that the difference between the two output activations $\Delta = \sigma - \sigma'$ is positive when $y = 1$, and negative when $y = 0$. Ensuring this would take us closer to the actual objective, which is Kendall's τ. One possible way to do this is to introduce a task-specific cost function that penalizes the disagreements similarly to the way Kendall's τ does.[4] In particular, we define a new *Kendall cost* as follows:

$$J_\theta = -\sum_n y_n \,\mathrm{sig}(-\gamma\Delta_n) + (1 - y_n)\,\mathrm{sig}(\gamma\Delta_n) \tag{6}$$

where we use the sigmoid function sig as a differentiable approximation to the step function.

The above cost function penalizes disconcordances, i.e., cases where (*i*) $y = 1$ but $\Delta < 0$, or (*ii*) when $y = 0$ but $\Delta > 0$. However, we also need to make sure that we discourage *ties*. We do so by adding a zero-mean Gaussian regularization term $\exp(-\beta\Delta^2/2)$ that penalizes the value of Δ getting close to zero. Note that the specific values for γ and β are not really important, as long as they are large. In particular, in our experiments, we used $\gamma = \beta = 100$.

Table 5 shows a comparison of the two cost functions: (*i*) the standard logistic cost, and (*ii*) our Kendall cost. We can see that using the Kendall cost enables effective learning, although it is eventually outperformed by the logistic cost. Our investigation revealed that this was due to a combination of slower convergence and poor initialization. Therefore, we further experimented with a setup where we first used the logistic cost to pretrain the neural network, and then we switched to the Kendall cost in order to perform some finer tuning. As we can see in Table 5 (last row), doing so yielded a sizable improvement over using the Kendall cost only; it also improved over using the logistic cost only.

[4]Other variations for ranking tasks are possible, e.g., (Yih et al., 2011).

7 Conclusions and Future Work

We have presented a novel framework for learning a tunable MT evaluation metric in a pairwise ranking setting, given pre-existing pairwise human preference judgments.

In particular, we used a neural network, where the input layer encodes lexical, syntactic and semantic information from the reference and the two translation hypotheses, which is efficiently compacted into relatively small embeddings. The network has a hidden layer, motivated by our intuition about the problem, which captures the interactions between the relevant input components. Unlike previously proposed kernel-based approaches, our framework allows us to do both training and inference efficiently. Moreover, we have shown that it can be trained to optimize a task-specific cost function, which is more appropriate for the pairwise MT evaluation setting.

The evaluation results have shown that our NN model yields state-of-the-art results when using lexical, syntactic and semantic features (the latter two based on compact embeddings). Moreover, we have shown that the contribution of the different information sources is additive, thus demonstrating that the framework can effectively integrate complementary information. Furthermore, the framework is flexible enough to exploit different granularities of features such as n-gram matches and other components of BLEU (which individually work better than using the aggregated BLEU score). Finally, we have presented evidence suggesting that using the pairwise hidden layers is advantageous over simpler flat models.

In future work, we would like to experiment with an extension that allows for multiple references. We further plan to incorporate features from the *source* sentence. We believe that our framework can support learning similarities between the two translations and the source, for an improved MT evaluation. Variations of this architecture might be useful for related tasks such as Quality Estimation and hypothesis re-ranking for Machine Translation, where no references are available.

Other aspects worth studying as a complement to the present work include (*i*) the impact of the quality of the syntactic analysis (translations are often just a "word salad"), (*ii*) differences across language pairs, and (*iii*) the relevance of the domain the semantic representations are trained on.

References

Joshua Albrecht and Rebecca Hwa. 2008. Regression for machine translation evaluation at the sentence level. *Machine Translation*, 22(1-2):1–27.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, ACL '14, pages 238–247, Baltimore, Maryland, USA.

Yoshua Bengio and Xavier Glorot. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of AI & Statistics 2010*, volume 9, pages 249–256, Chia Laguna Resort, Sardinia, Italy.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. 2010. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference*, SciPy '10, Austin, Texas.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, WMT '14, pages 12–58, Baltimore, Maryland, USA.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2007. (Meta-) evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, WMT '07, pages 136–158, Prague, Czech Republic.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar Zaidan. 2011. Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT '11, pages 22–64, Edinburgh, Scotland.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 Workshop on Statistical Machine Translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, WMT '12, pages 10–51, Montréal, Canada.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Elisabet Comelles, Jesús Giménez, Lluís Màrquez, Irene Castellón, and Victoria Arranz. 2010. Document-level automatic MT evaluation based on discourse representations. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10, pages 333–338, Uppsala, Sweden.

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*, WMT '11, pages 85–91, Edinburgh, Scotland.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, ACL '14, pages 1370–1380, Baltimore, Maryland, USA.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, pages 138–145, San Francisco, California, USA. Morgan Kaufmann Publishers.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159.

Kevin Duh. 2008. Ranking vs. regression in machine translation evaluation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, WMT '08, pages 191–194, Columbus, Ohio, USA.

Jesús Giménez and Lluís Màrquez. 2007. Linguistic features for automatic evaluation of heterogenous MT systems. In *Proceedings of the Second Workshop on Statistical Machine Translation*, WMT '07, pages 256–264, Prague, Czech Republic.

Francisco Guzmán, Shafiq Joty, Lluís Màrquez, Alessandro Moschitti, Preslav Nakov, and Massimo Nicosia. 2014a. Learning to differentiate better from worse translations. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, EMNLP '14, pages 214–220, Doha, Qatar.

Francisco Guzmán, Shafiq Joty, Lluís Màrquez, and Preslav Nakov. 2014b. Using discourse structure improves machine translation evaluation. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics*, ACL '14, pages 687–698, Baltimore, Maryland, USA.

Shafiq Joty, Francisco Guzmán, Lluís Màrquez, and Preslav Nakov. 2014. DiscoTK: Using discourse structure for machine translation evaluation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, WMT '14, pages 402–408, Baltimore, Maryland, USA.

Alex Kulesza and Stuart M. Shieber. 2004. A learning approach to improving sentence-level MT evaluation. In *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation*.

Alon Lavie and Michael Denkowski. 2009. The METEOR metric for automatic evaluation of machine translation. *Machine Translation*, 23(2–3):105–115.

Ding Liu and Daniel Gildea. 2005. Syntactic features for evaluation of machine translation. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 25–32, Ann Arbor, Michigan, USA.

Chi-kiu Lo, Anand Karthik Tumuluru, and Dekai Wu. 2012. Fully automatic semantic MT evaluation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, WMT '12, pages 243–252, Montréal, Canada.

Matouš Macháček and Ondřej Bojar. 2013. Results of the WMT13 metrics shared task. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, WMT '13, pages 45–51, Sofia, Bulgaria.

Matouš Macháček and Ondřej Bojar. 2014. Results of the WMT14 metrics shared task. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, WMT '14, pages 293–301, Baltimore, Maryland, USA.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *11th Annual Conference of the International Speech Communication Association*, pages 1045–1048, Makuhari, Chiba, Japan.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013a. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, NIPS '13, pages 3111–3119. Lake Tahoe, California, USA.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '13, pages 746–751, Atlanta, Georgia, USA.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1439.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meting of the Association for Computational Linguistics*, ACL '02, pages 311–318, Philadelphia, Pennsylvania, USA.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '14, pages 1532–1543, Doha, Qatar.

Maja Popović and Hermann Ney. 2007. Word error rates: Decomposition over POS classes and applications for error analysis. In *Proceedings of the Second Workshop on Statistical Machine Translation*, WMT '07, pages 48–55, Prague, Czech Republic.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Biennial Conference of the Association for Machine Translation in the Americas*, AMTA '06, Cambridge, Massachusetts, USA.

Richard Socher, John Bauer, Christopher D. Manning, and Ng Andrew Y. 2013a. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL '13, pages 455–465, Sofia, Bulgaria.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013b. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, EMNLP '13, pages 1631–1642, Seattle, Washington, USA.

Xingyi Song and Trevor Cohn. 2011. Regression and ranking based optimisation for sentence-level MT evaluation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT '11, pages 123–129, Edinburgh, Scotland.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the Neural Information Processing Systems*, NIPS '14, Montreal, Canada.

Billy Wong and Chunyu Kit. 2012. Extending machine translation evaluation metrics with lexical cohesion to document level. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1060–1068, Jeju Island, Korea.

Wen-tau Yih, Kristina Toutanova, John C. Platt, and Christopher Meek. 2011. Learning discriminative projections for text similarity measures. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, CoNLL '11, pages 247–256, Portland, Oregon, USA.