Semantically Aware Objective Functions for Referring Expression Tasks

Jake Bass [*] [1] **Ishaan Jhaveri** [*] [1] **Bennett Norman** [*] [1] **Jack Tregurtha** [*] [1]

Abstract

We propose a set of objective functions that can be used for any text generation task. These functions utilize recent work on high-quality word embeddings to gain an awareness of semantics. We show their ability to improve performance on the Referring Expression task.

1. Introduction

Recent work has shown great success on a myriad of text generation tasks. From image captioning to machine translation, recent advances in deep learning for Natural Language Processing have allowed us to realize impressive performance improvements on many of these tasks. However, relatively little attention has been paid to the objective function used to optimize these networks. In fact, current approaches take a rather naïve approach that: (1) tends to overfit to ground truth sequences, (2) does not take into account that for most language generation tasks, there are many equally plausible correct solutions, and (3) does not take into account semantic information. We propose a set of objective functions derived from Word Mover's Distance (Kusner et al., 2015) to ameliorate each of these issues. Word Mover's Distance proposes a distance function between text documents, and is based on recent work that learns semantically meaning representations for words (Mikolov et al., 2013; Pennington et al., 2014). We apply our set of objective functions to the Referring Expression task and achieve substantial performance improvements.

2. Related Work and Baseline

Generating natural language is currently a popular topic in machine learning. There is also an increasingly popular set of multi-modal tasks that require joint understanding of vision and language. One of the most popular of these tasks is Image Captioning. The image captioning task takes images as input and produces captions describing the images as output. However, for any given image, there is a incredibly large space of captions that could be considered accurate descriptions of the image. Consequently, judging the quality of these captions is a very difficult and yet unsolved problem. This ambiguity motivated the creation of the Referring Expression task.

We will now introduce the Referring Expression task, popular approaches to the task, and other relevant related work. We will also go into depth on our baseline model and how it derives from current state of the art work, and on the popular Cross Entropy Loss.

2.1. Related Work

Referring Expression Task. The Referring Expression Task is motivated by the ambiguity of image captioning. The generation task is, given an image and a region of the image containing an object, produce an expression that umabiguously describes the object. The comprehension task is, given an expression and an image, highlight the region of the image that contains the object that the expression describes. This paper discusses the generation task.

Prior Work. In 2014, (Kazemzadeh et al., 2014) introduced the first large-scale dataset with real-world images for referring expressions. (Mao et al., 2015) followed up on that work with one of the first deep learning approaches to the generation task. It introduced a CNN-LSTM baseline frequently referenced in future work. (Yu et al., 2016a) builds upon the generation baseline established by Mao by encoding more visual context into the visual features. The model also penalizes ambiguous expressions. (Yu et al., 2016b) expands this work with a joint model that incorporates generation, comprehension, and reinforcement learning components into one large model. These components are trained jointly. As far as we know, this joint model is the current state of the art.

Word2vec/GloVe: (Mikolov et al., 2013) created the first high-quality semantically-aware word embeddings, called *word2vec*. They did so by learning word embedding vectors such that similar vectors in the embedded space share similar word contexts in natural language. *GloVe* also learns semantically aware embeddings, but does so via the co-occurence probabilities of words instead of their local

[*] Equal contribution [1] Cornell University. Correspondence to: Jake Bass <jab783@cornell.edu>, Ishaan Jhaveri <iaj8@cornell.edu>, Bennett Norman <bdn29@cornell.edu>, Jack Tregurtha <jt548@cornell.edu>.

contexts (Pennington et al., 2014).

Word Mover's Distance: Word Mover's Distance (WMD) is a distance function between text documents. WMD measures dissimilarity between the documents by utizilizing semantically meaningful word embeddings and looking at the minimum distance to transform one document, represented as a collection of its word embeddings, into another. (Kusner et al., 2015)

Gumbel-Softmax: The Gumbel-Softmax trick allows us to use the popular reparameterization trick on a discrete distribution, and thus sample from a discrete distribution while maintaining differentiability (Jang et al., 2016).

2.2. Baseline

Our baseline attempts to replicate the CNN-LSTM baseline introduced in (Mao et al., 2015) and used in (Yu et al., 2016a;b). Our model takes a image, the region of the image specified by the object of interest's bounding box, and the bounding box coordinates as input. We then feed our image and region through a ResNet-152 CNN with its parameters frozen, and then through a fully-connected neural network linear layer to produce 256-dimensional feature vectors representing the image and region respectively (He et al., 2015). Both vectors are then concatenated with a 4-tuple that represents our bounding box: $[\frac{x_{tl}}{W}, \frac{y_{tl}}{H}, \frac{x_{br}}{W}, \frac{y_{br}}{H}]$ where (x_{tl}, y_{tl}) and (x_{br}, y_{br}) are the coordinates of the top left and bottom right corners of the object bounding box, and H and W are height and width of the bounding box. This 516-dimensional feature vector is concatenated with the embedding of the ground truth expression and fed into an LSTM with a hidden size of 512. The model is trained with Cross Entropy loss.

2.3. Cross Entropy Loss

Cross Entropy Loss is a popular objective function for multi-class classification problems. It seeks to minimize the difference between the predicted probability distribution q and the ground truth distribution p for all data points. For a single prediction, the cross entropy loss is:

$$H(p, q) = -\sum_{j}^{C} p_j \log q_j$$

where q_j is the predicted probability of the j^{th} token, p_j is the ground truth probability of the j^{th} token, and C is the size of (total number of tokens in) the vocabulary. H is then averaged across all predictions.

In almost all cases, for the ground truth distribution p, the correct token is assigned probability 1 and *every other token* is assigned probability 0. Consequently, minimizing Cross Entropy loss is equivalent to maximizing the log *predicted* probability of the ground truth token. This leads to a distribution of possible generated expressions for a given image and bounding box that looks like **Figure 1**.

There are two major issues with objective functions that produce this kind of distrbution: (1) they cause the model to overfit to the ground truth expression, reducing generalizability to unseen data at test time, and (2) the objective function has no awareness of **semantic meaning**; it simply looks for token overlap between the ground truth and predicted distributions. Therefore, expressions that are semantically very similar to the ground truth but do not share many tokens with it will be penalized, and expressions that share many tokens with the ground truth but have entirely different meaning will be rewarded.

We propose that these issues can be combated by using a semantically aware objective function.

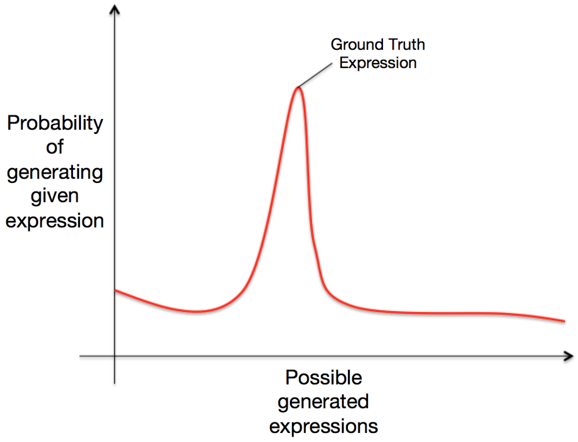


Figure 1. The distribution of expressions generated by a model trained with Cross Entropy. At the very peak of the distribution is the ground truth expression, which has the highest probability of being generated. On either side of the peak are expressions that have many tokens in common with the ground truth expression, and consequently they have relatively high probabilities of being generated. Finally, most expressions share few or no tokens in common with the ground truth expression, and so they have low probabilities.

3. Methodology

This section introduces our notation and then describes our methodology for incorporating Word Mover's Distance, Word Centroid Distance, and the newly proposed Word Embedding Distance as objective functions.

3.1. Notation and Notes

At each training step, a ground truth expression and corresponding image is chosen. (Note: in practice a batch of data is selected, but for the sake of notational and explanatory simplicity, we will think of each batch being of size 1.) Let N be the total number of tokens in the given ground truth expression. For each ground truth token, we will make a prediction. However, instead of predicting a single token, we will actually predict a distribution over all tokens in our vocabulary. So, for each of the N ground truth tokens t_i, a distribution D_i is predicted. Let D_{ij} denote the probability of the j^{th} token in the vocabulary in the i^{th} predicted distribution. Also, let $emb[token]$ denote the word embedding for the given token.

Additionally, note that during training, a distribution is predicted for each ground truth token. This implies that each predicted expression is the same length as its corresponding ground truth expression. However, a special <end> token is added to the vocabulary and to the end of every ground truth expression. During testing, our model predicts tokens sequentially until an <end> token is predicted. Thus, our model can predict expressions of varied lengths during testing.

Additionally, note that we use *GloVe* embeddings for all of our experiments.

3.2. Word Mover's Distance with the Gumbel-Softmax Trick

The first semantically aware loss function we tried was Word Mover's Distance (WMD). As discussed previously, expressions are not normally generated during training; instead, a distribution over all possible tokens in the vocabulary is predicted (D_i above). However, WMD is a calculation between two expressions. More specifically, it is a calculation between the embeddings for each token in the expression. To use WMD, we needed to convert each predicted distribution to an embedding representing that distribution. To do so, we decided simply to take the most probable token from each predicted distribution D_i and use this token as our prediction. This effectively yields generated expressions. We then compute the WMD score between these generated expressions and the ground truth expressions.

However, the operation of taking the most likely token from each predicted distribution is non-differentiable, and we would not have been able to backpropagate and update the model parameters. To get around this, we used the Gumbel Softmax Re-Parametrization Trick to transform the predicted distribution for each token into a continuous distribution through the addition of Gaussian noise (Jang et al., 2016) and thus make sampling differentiable.

However, the expressions produced by a model trained with just WMD were utter nonsense. They are shown in figure 5 in the appendix. They had a lot of repeating words, and absolutely no idea of the given image or of position of each word in the expression. This model produced scores of roughly 0.0 on every evaluation metric. After substantial trouble debugging what was going wrong, we decided to simplify our model.

3.3. Expected Word Embeddings and Word Centroid Distance

We make two dramatic changes to our previous approach that simply used WMD as an objective function. The first revolves around the way we move from our predicted distribution D_i to predicted word and then predicted word embedding. Instead of using the Gumbel trick to take the most likely word for the predicted distribution and then using that word's embedding, we directly compute the expected word embedding for each prediction. So, for each ground truth token t_i, we compute the *expected word embedding* e_i as a weighted sum of the probability of each word and that word's embedding:

$$e_i = \sum_{j}^{C} D_{ij} \cdot emb[vocab[j]]$$

Calculating the expected word embedding using the entire predicted distribution has a few important effects: (1) we make use of all the information the model has (instead of throwing away most of it) (2) we help the model calibrate probabilities for each word; the relative probabilities now matter, instead of just the size of the probability that corresponds to ground truth (3) we can simply use the expected word embedding for each prediction, so we no longer require the Gumbel trick.

Our second change was switching from WMD to Word Centroid Distance (WCD). WCD similarly makes use of semantically aware word embeddings, but does so in a much simpler fashion. To compare two expressions, WCD computes the average embedding of each expression and then compares each expression's average embedding, or centroid, using any standard distance metric. Additionally, we experiment with a similar computation we introduce called Word Embedding Distance (WED), where for each token in the given expression we compute the distance between the ground truth embedding and the expected word embedding, and then we average every distance across all predictions in an expression.

These changes yield promising results (see table 1). We experiment using both WCD and WED as our objective function. In addition to training models using WCD and WED from scratch, we also experiment with models that are initialized with the parameters of our trained baseline model and then trained with WCD or WED from that point. We refer to this practice as training ”on top of” the baseline model. Additionally, we experiment with hybrid objective functions that combine Cross Entropy with one of our semantically aware loss functions. However, the two objective functions have drastically different scales. We combat this by keeping a running average of each respective loss and weighting our semantically aware loss by the ratio of running averages. Thus, our hybrid objective function is defined as:

$$L_{hybrid} = \gamma \cdot L_{CE} + (1 - \gamma) \cdot (L_{SA} \cdot \frac{L_{CE\_AVG}}{L_{SA\_AVG}})$$

where L_{CE} is the cross entropy loss from that training step, L_{SA} is the loss of one of our semantically aware objective functions (WMD, WCD, or WED) from that step, L_{CE_AVG} and L_{SA_AVG} are the running averages of the respective losses up to and including the current step, and γ is a hyperparameter scaling factor.

3.4. Positional Encoding

While our class of semantically aware objective functions appear promising, they do have one major shortcoming: neither WCD nor WMD account for positional information in comparing predicted and ground truth expressions. We account for this lack of positional encoding in two ways: either (1) training on top of a model pretrained with a positionally-aware loss, or (2) training with a hybrid objective function where one component is positionally-aware.

4. Results

In table 1, **Objective Function** refers to whether the given model used Word Embedding Distance (WED) or Word Centroid Distance (WCD) as its objective function. **Pretrained/From Scratch** refers to whether the given model was trained from scratch with randomly initialized parameters, or whether it was trained on top of a model that was trained with cross entropy. **Hybrid/Alone** refers to whether the given model used purely the objective function specified in the first column, or whether it used the hybrid objective function defined in section **3.3** with cross entropy and the listed objective function as its components. In this case, the number after “Hybrid” is the value of γ.

Models annotated with a † achieved best results with early stopping. Models annotated with a ‡ deteriorated especially quickly as training proceeded. The results shown for these models are after one epoch.

All of our models achieve some improvement over our baseline. At the time of writing this report, our baseline was achieving significantly lower than scores than the baseline of (Yu et al., 2016a) on all of the evaluation metrics. However, since writing this report, we have been able to recreate their baseline’s scores almost exactly by using early stopping on our baseline. We are now running the experiments that use this improved baseline as the pretrained model to see if we still improve our scores on the metrics even above this increased baseline.

4.1. Qualitative Analysis

Many of our semantically aware objective functions are able to achieve higher scores on the evaluation metrics than our cross entropy baseline. While these results appear promising, we decided to investigate exactly where these improvements were coming from. We hoped this would help us understand which parts of our hypothesis were supported. The appendix shows a few of the instances for which the BLEU-1 score improved the most from the baseline to our best model (line 3 in **table 1**). It seems from these results that the models trained on semantically aware objectives still predict expressions that are close to the ground truth expressions of the test set. However, it is unsurprising that this is what we observe given that we are looking at expressions with the most improved scores on a metric that captures N-gram overlap. The fact that these models still do better on unseen data means that they overfit less than the cross entropy model. While this qualitative performance seems promising, we plan on more thoroughly investigating our results with respect to improvement both on other metrics and on our semantically aware objective functions.

5. Future Work

Improve Baseline: As is clear from Table 1 in the Results section, our baseline is still lower than the baseline of (Yu et al., 2016a) on all metrics. We want to bring our baseline up to par with theirs and then show that using our objective function still gives an increase in scores on these metrics. We hope that this will show that the improvements generated by using our objective function are not already captured by another part of their quite complex state of the art model. We have access to the (Yu et al., 2016a) baseline code and are in the process of adding all the features to our baseline from theirs that we currently have not added.

Direct Extensions There are many direct extensions to the experiments we have completed that we would like to investigate. Here are a few:

- **Different distance metrics.** We currently use L1 loss for the distance calculation between embeddings for WCD and WED. We would like to try other losses such as L2, cosine, or other Minkowski distances.

- **WMD.** Since we now calculate expected word embeddings, we can use them for a WMD calculation without requiring Gumbel. Although WMD does not encode word order, we would like to try using it with our expected word embeddings either as a hybrid objec-

Table 1. Results for selected experiments with WCD and WED on the TestA split of the RefCOCO dataset

Objective Function	Pretrained / Scratch	Hybrid / Alone	BLEU-1	BLEU-2	ROUGE-L	METEOR	BLEU-3	BLEU-4	CIDEr
(Yu et al., 2016a) Baseline			0.477	0.290	0.413	0.173	-	-	-
Our Baseline			0.440	0.257	0.393	0.173	0.149	0.084	0.641
WED	Pretrained	Alone	**0.487**	**0.303**	**0.423**	**0.182**	**0.186**	**0.105**	**0.697**
WED‡	Pretrained	Hybrid (0.5)	0.461	0.270	0.394	0.172	0.155	0.087	0.651
WED†	Scratch	Hybrid (0.5)	0.478	0.284	0.402	0.172	0.170	0.100	0.653
WCD†	Pretrained	Alone	0.019	0.005	0.046	0.037	0.001	0.000	0.006
WCD‡	Pretrained	Hybrid (0.5)	0.455	0.264	0.402	0.177	0.151	0.092	0.664
WCD†	Scratch	Hybrid (0.75)	0.469	0.278	0.406	0.176	0.169	0.078	0.670

† Best results occur with early stopping

‡ Results shown are after one epoch; model performance reduces drammatically the longer it trains

tive function or on its own on top of a model trained with a positionally aware objective function, as laid out in 3.4.

- **Normalized embeddings.** All of our semantically aware objective functions rely heavily on word embeddings. However, word embeddings can be of substantially different magnitudes. We would like to see how they perform if the embeddings are normalized. Additionally, we would like to try out using *word2vec* or other embeddings in place of *GloVe* embeddings.
- **Handling of missing words.** Currently, more than 10% of the words in the RefCOCO training data do not have *GloVe* embeddings. We handle this by randomly initializing their embedding at the start of training. However, we would like to experiment with more elegant modes of handling these missing words.
- **Hyperparameter optimization.** For the sake of time and consistency across models, we have not played with optimizing many of our hyperparameters. We have tried only a few values for γ for our hybrid loss, and we have kept our learning rate constant for all models. We would like to experiment with a wider range of values for both.
- **LSTM Initialization and Special Tokens.** Currently, our LSTM is initialized randomly and the visual features are fed in as the first input (and a start token as the second input). This is an unfortunate relic of the codebase we inherited, and the fact that we did not discover this until too far into our experiments to rerun them all. We would like to initialize the LSTM with our visual features, thus removing the need for a start token in our ground truth. We think this may improve performance on our objective functions, which were dramatically affected by the frequent appearances of start and end tokens. (We have already removed start and end tokens from consideration for WCD (as it produced nonsensical results before removing them), but this likely limits its use to a component of a hybrid objective function.) We would also like to experiment with removing special tokens altogether from WED.

Other Domains: There is nothing about our class of objective functions that limits them to the Referring Expression task. In fact, they could be applied to any text generation task. We would like to see how they perform in different domains, where different standard objective functions than Cross Entropy are traditionally used. Planned domains for exploration are image captioning, machine translation, summarization, visual dialogue system, and visual question answering.

Postprocessing Word Embeddings: (Mu et al., 2017) demonstrate methods of postprocessing word embeddings to increase the amount of context-specific meaning encoded in the embeddings. We would like to see if these postprocessing operations on our embeddings improve our performance.

Reinforcement Learning: (Liu et al., 2016) greatly improves evaluation scores of an image captioning model by first training a CNN-LSTM using cross entropy as an objective and then optimizing for an evaluation metric using policy gradient and Monte Carlo rollouts. Optimizing for a combination of CIDEr and SPICE achieved results that were strongly preferred by human evaluators. (Kilickaya et al., 2016) showed the viability of WMD as an automatic evaluation metric for Image Captioning. The combination of these two papers makes us think using the policy gradient algorithm (Liu et al., 2016) optimized for WMD could be a good idea. However, we are unsure if a policy gradient would be better than using WCD as a loss.

Positional Encoding: Currently, we rely on Cross Entropy, either through pretraining or a hybrid loss, to encode position. We think that there may be more elegant ways to obtain positional awareness, but we are not sure which directions are the most promising.

References

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition, 2015.

Jang, Eric, Gu, Shixiang, and Poole, Ben. Categorical reparameterization with gumbel-softmax, 2016.

Kazemzadeh, Sahar, Ordonez, Vicente, Matten, Mark, and Berg, Tamara L. Referit game: Referring to objects in photographs of natural scenes, 2014.

Kilickaya, Mert, Erdem, Aykut, Ikizler-Cinbis, Nazli, and Erdem, Erkut. Re-evaluating automatic metrics for image captioning, 2016.

Kusner, M. J., Sun, Y., Kolkin, N. I., and Weinberger, K. Q. From word embeddings to document distances. In *ICML*, 2015.

Liu, Siqi, Zhu, Zhenhai, Ye, Ning, Guadarrama, Sergio, and Murphy, Kevin. Improved image captioning via policy gradient optimization of spider, 2016.

Mao, Junhua, Huang, Jonathan, Toshev, Alexander, Camburu, Oana, Yuille, Alan, and Murphy, Kevin. Generation and comprehension of unambiguous object descriptions, 2015.

Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg, and Dean, Jeffrey. Distributed representations of words and phrases and their compositionality, 2013.

Mu, Jiaqi, Bhat, Suma, and Viswanath, Pramod. Representing sentences as low-rank subspaces, 2017.

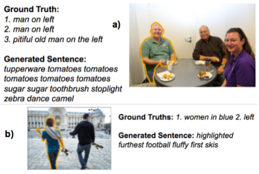
Pennington, Jeffrey, Socher, Richard, and Manning, Christopher D. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

Yu, Licheng, Poirson, Patrick, Yang, Shan, Berg, Alexander C., and Berg, Tamara L. Modeling context in referring expressions, 2016a.

Yu, Licheng, Tan, Hao, Bansal, Mohit, and Berg, Tamara L. A joint speaker-listener-reinforcer model for referring expressions, 2016b.

Appendix: Qualitative Results

Example Nonsensical Results from the First WMD-Gumbel Test



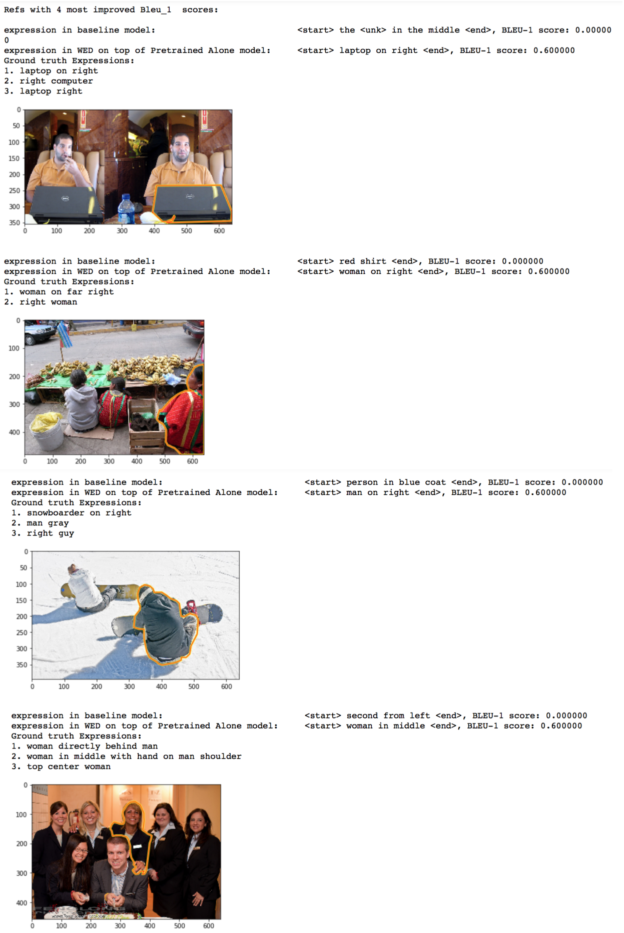
Ground Truth:
1. man on left
2. man on left
3. pitiful old man on the left

Generated Sentence:
tupperware tomatoes tomatoes tomatoes tomatoes tomatoes sugar sugar toothbrush stoplight zebra dance camel

Ground Truths: *1. women in blue 2. left*

Generated Sentence: *highlighted furthest football fluffy first skis*

Expressions that Improved Most From our Baseline to our Best Model



```
Refs with 4 most improved Bleu_1  scores:

expression in baseline model:                         <start> the <unk> in the middle <end>, BLEU-1 score: 0.00000
0
expression in WED on top of Pretrained Alone model:   <start> laptop on right <end>, BLEU-1 score: 0.600000
Ground truth Expressions:
1. laptop on right
2. right computer
3. laptop right
```

```
expression in baseline model:                         <start> red shirt <end>, BLEU-1 score: 0.000000
expression in WED on top of Pretrained Alone model:   <start> woman on right <end>, BLEU-1 score: 0.600000
Ground truth Expressions:
1. woman on far right
2. right woman
```

```
expression in baseline model:                         <start> person in blue coat <end>, BLEU-1 score: 0.000000
expression in WED on top of Pretrained Alone model:   <start> man on right <end>, BLEU-1 score: 0.600000
Ground truth Expressions:
1. snowboarder on right
2. man gray
3. right guy
```

```
expression in baseline model:                         <start> second from left <end>, BLEU-1 score: 0.000000
expression in WED on top of Pretrained Alone model:   <start> woman in middle <end>, BLEU-1 score: 0.600000
Ground truth Expressions:
1. woman directly behind man
2. woman in middle with hand on man shoulder
3. top center woman
```