
A Discussion of Semantically Aware Objective Functions for Language Generation Tasks

Jake Bass ^{*1} Ishaan Jhaveri ^{*1} Bennett Norman ^{*1}

Abstract

In a previous work, we proposed a set of objective functions that can be used for any text generation task. These functions utilize recent progress in high-quality word embeddings to gain an awareness of semantics. In a holistic review of our past and current work, we explore their efficacy and their applicability to the Referring Expression task and Language Generation tasks in general.

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 techniques 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 when comparing words. We previously proposed a set of objective functions to ameliorate each of these issues. In this work, we will briefly cover our prior creations and their results, we will introduce a new type of semantically-aware objective function motivated by our prior work, and we discuss how our findings should shape future exploration.

2. Background Information

We previously proposed a set of semantically-aware objective functions and applied them to the Referring Expression task. We will now explain these functions and the Referring

Expression task.

Referring Expression Task: Referring expressions are phrases that identify an object in an image (e.g. *lady in red* or *elephant on the left*). The Referring Expression Generation (REG) task is: given an image and a region of the image containing an object, produce an expression that unambiguously describes the object. The Referring Expression Comprehension task is: given an image and an expression, highlight the region of the image that contains the object that the expression describes. This paper discusses the generation task.

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

Word Centroid Distance: Word Centroid Distance (WCD) (Kusner et al., 2015) is also a distance function between text documents. It is similar to WMD, but is faster to calculate. Instead of calculating the cost to transform one document to another, WCD computes a centroid for each document, which is the average word embedding for all words in the document weighted by each word’s frequency. The WCD between two documents is defined as the distance between each document’s respective centroid.

Word Embedding Distance: Word Embedding Distance (WED) is also a distance metric between two documents. It was created in our previous work. WED takes two same-length documents and computes the distance between the first word of each, the second word of each, etc. WED is the average of each of these distances.

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

^{*}Equal contribution ¹Cornell University. Correspondence to: Jake Bass <jab783@cornell.edu>, Ishaan Jhaveri <iaj8@cornell.edu>, Bennett Norman <bdn29@cornell.edu>.

3. Our Prior Work

We previously explored the viability of using WMD, WCD, and WED as objective functions in our work [Semantically Aware Objective Functions for Referring Expression Tasks](#) (Bass et al.). We will now summarize that work, and explain how it has informed more recent experimentation.

3.1. Document Distance Metrics as Objective Functions

WMD and WCD were originally formulated as metrics for text document similarity. However, it is easy to think of them as objective functions for Referring Expression Generation (REG), or any other Language Generation task. While training a model, we simply need to consider the ground truth expression and the generated expression to be “documents”. However, doing so raises two major issues: differentiability and positional encoding.

3.1.1. DIFFERENTIABILITY

Most models, ours included, do not produce words during training; instead, they produce a categorical distribution over the vocabulary. WMD, WCD, and WED all require words - or more specifically, word embeddings - as input. The simple solution to produce words to be input would be to sample from the predicted categorical distribution, as is done at test time, and then take the sampled words’ respective embedding. However, the sampling operation is not differentiable. Thus begs the question: how do we convert a categorical distribution over our vocabulary into a word embedding?

We use two solutions. The first solution is to use the Gumbel-Softmax trick, as described in section 2. The second solution is to use what we call an **expected embedding**. For a predicted categorical distribution over a vocabulary, the expected embedding is the weighted average of the embedding of every word in the vocabulary, where each word embedding’s weight is the probability of that word in the categorical distribution. Thus, we are able to transform the distribution into an embedding through actions that are all differentiable.

3.1.2. POSITIONAL ENCODING

The second major issue we encountered was with positional encoding. WMD and WCD have no sense of where words occur in a sentence or document. While this is not an issue for evaluating existing text (e.g. in assessing document similarity), it poses a large challenge when learning to generate text. We came up with two solutions to this problem. The first is to create a hybrid loss that is a weighted combination between Cross Entropy (a standard loss for Language Generation tasks) and one of our objective functions. The second solution was to first train a model with

Cross Entropy, and then refine it by training more with one of our objective functions. We refer to this condition as training on top of a pretrained model, or pretraining.

3.2. Results of Prior Work

We achieved modest performance improvements over our baseline in many of our experimental conditions. However, there were also many where generated expressions degenerated quickly into nonsense. In particular, models trained with WED or WCD that used either a hybridized loss or pretraining showed at least modest improvements over our baseline in many experimental conditions. However, all models trained with WMD - regardless of whether we used Gumbel vs. Expected Embeddings, or pretraining vs. hybridized loss - produced nonsensical expressions.

3.3. New Direction for This Work

Despite moderate success with our past work, and that there was much more to explore, we decided to move in a new direction. We thought that there must be a smarter, simpler way to incorporate semantic information into an objective function. We sought a method that was:

- **more robust** - it could be trained from scratch, didn’t require hybridization and the accompanying hyperparameter optimization or pretraining, and naturally encoded position
- **simpler** - didn’t require the complexity of Gumbel or pretraining for expected embeddings
- **more interpretable** - easier to understand what was working and what wasn’t

In pursuit of these goals, we created the modified Cross Entropy (mCE) objective function. We will now discuss various formulations of mCE and the experiments we conducted with it.

4. Experiments

4.1. Baseline Architecture

In following our new direction, we switched from using our experimental architecture to the current state of the art (SOTA) architecture. At this time, the state of the art for Referring Expression Generation is a CNN-LSTM architecture from (Yu et al., 2016). They incorporate a Referring Expression Comprehension module and Reinforcement Learning module to the CNN-LSTM.

4.2. Modified Cross Entropy Loss

In our prior work (Bass et al.), we extensively cover Cross Entropy loss and its shortcomings. See section 2.3 of our

previous paper for more details. For a single token, Cross Entropy Loss is defined as:

$$-\sum_j^C p_j \log q_j \quad (1)$$

where q_j is the predicted probability of the j^{th} token in the vocabulary, p_j is the ground truth probability of the j^{th} token in the vocabulary, and C is the size of (total number of tokens in) the vocabulary. This value is then averaged across all tokens in a given expression to get that expression’s loss, and then across predicted expressions to get a total loss.

For each possible ground truth label (i.e. for each word in the vocabulary), there exists p , a distribution over the vocabulary. We will call the p the ground truth distribution. This distribution can be thought of as a vector of length C , which we will call the ground truth distribution vector.

The k^{th} entry of the ground truth distribution vector for a given ground truth label gtl can be interpreted as such: what is the probability that the k^{th} token in the vocabulary should be the ground truth label, given that gtl actually is the ground truth label?

For Cross Entropy, the k^{th} token (i.e. the correct token) is assigned probability 1 and every other token is assigned probability 0. This is the most intuitive approach. However, there are some drawbacks.

Figure 1.



Take the image above, and say we are asked to describe the man. Say the ground truth expression was “man on right”, and we are looking at the first token of the expression: “man”. The first drawback is that the ground truth label is not perfectly assigned for the given image, as there are often other good options to describe an object. In this case, the words “guy” or “gentleman” may describe him just as well. Additionally, all other tokens in the vocabulary are not equally unlikely. The word “gentleman” is clearly

a more accurate description of the man than the word “elephant”. However, they will both be given a probability of 0.

With mCE, we seek to address these issues. Instead of assigning all probability mass to a single token, thus indicating there is only one potentially valid prediction, we create a distribution over all potential tokens. This is the core intuition behind mCE.

4.3. mCE Formulations

While we have described what mCE is, we have not explicitly defined it, or explained how the ground truth distribution is created. We define our ground truth matrix as a C by C matrix where each row is the ground truth distribution vector for a given ground truth label. Thus, the j^{th} entry in the i^{th} row of the matrix gives the probability of the j^{th} word in the vocabulary in the ground truth distribution of the i^{th} word in the vocabulary.

We define a formulation of mCE by how we populate this ground truth matrix. Notably, the ground truth matrix for Cross Entropy is the identity matrix. For mCE, we seek to set a ground truth matrix that incorporates semantic information when defining each ground truth distribution.

Now we will explain each of the formulations of mCE that we experimented with and the intuitions behind them. We present the results of the corresponding experiments in the Results section.

1. Softmax over Pairwise Cosine Similarity Matrix:

One way of incorporating semantic information in the ground truth distribution for a given ground truth label is to assign a probability to each token in the vocabulary that is a function of the similarity between that token’s embedding and the ground truth label token’s embedding. We tried Euclidean Distance and Cosine Similarity as the similarity metric and found empirically that Cosine Similarity works better. Consequently, we defined the ground truth matrix such that the i^{th} ground truth distribution was:

$$p_i = \sigma(-\lambda \cdot \text{cossim}(t_j, gtl)) \quad (2)$$

where σ is the softmax function, λ is an empirically set parameter, cossim is the cosine similarity between two words’ embeddings, t_j is the j^{th} token in the ground truth distribution, and gtl is the ground truth label for the i^{th} row of the ground truth matrix.

2. K-Nearest Neighbors with Similarity Score Threshold:

We found that the above approach assigns a lot of probability mass to tokens that have very low similarity scores with the ground truth label token, since there are so many of them (over 2000).

We also found that some tokens have very few other tokens in the vocabulary they are similar to, or none at all. So we employed two methods of controlling for these observations.

In each ground truth distribution vector, we assigned non-zero values only to the tokens with the K highest similarity scores. This change prevents there from being thousand of tokens with small probabilities, thus reducing noise.

Secondly, we assigned non-zero values only to tokens with similarity scores over a given threshold. Consequently, when a ground truth label lacks many synonyms or words that could be substituted for it, we do not assign misfit words high probabilities.

For each ground truth distribution, we divided the probability mass equally among the up-to- K tokens that were determined to have non-zero probabilities in the above manner.

3. **Synonyms Derived from WordNet Synsets:** Princeton University’s WordNet (wordnet.princeton.edu) lexical database contains *synsets* for each word in English. For any word, a synset gives its synonyms. Given the KNN shortcomings we discuss in the Results section, we tried a formulation of mCE where, for a given ground truth distribution, we assign non-zero values only to the tokens in the vocabulary that occur in the ground truth label’s synset. To derive a given token’s distribution, we divided the probability mass equally among all the other tokens that were determined to have non-zero probabilities in the above manner. Note this approach marks a departure from probabilities derived in any way from similarities between word embeddings.
4. **Handpicked Synonyms of 200 most common tokens:** As described in the Results section, the synset synonyms suffered from some shortcomings. We decided then to try the simplest possible formulation of probabilities derived from synonyms. We determined the distribution for a given ground truth label as follows. If it was not one of the 200 most common tokens in the training data, its ground truth distribution was just the 1-hot vector used in Cross Entropy. If it was one of the 200 most common tokens, we manually selected only the highest quality synonyms from its synset. Once again, probability mass was split equally between all qualified tokens.
5. **Handpicked Antonyms of 200 most common tokens:** We decided instead of promoting semantic similarity in our ground truth distributions, to discourage semantic dissimilarity. We determined the distribution for a given ground truth label as follows. If it

was not one of the 200 most common tokens in the training data, its ground truth distribution was just the 1-hot vector used in Cross Entropy. If it was one of the 200 most common tokens, we manually selected only the highest quality antonyms from its WordNet antonym set. We then assigned a “probability” of 1 to the token itself, and a “probability” of -1 to all selected antonyms. While this technique does not strictly define a probability distribution, there is no reason that each ground truth distribution vector must actually encode a distribution.

Table 1. METEOR and CIDEr scores from our 5 classes of experiments.

Model Description	CIDEr	METEOR
Yu’s SOTA	1.064	0.236
Handpicked Synonyms	1.029	0.235
Synset Synonyms	0.49	0.203
mCE with partial softmax	1.085	0.236
KNN with Cosine Similarity	1.056	0.234
Antonyms	0.052	0.014

5. Results

Table 1 includes the best METEOR and CIDEr scores of our 5 formulations of mCE. Additionally, Yu’s state of the art scores are included as a baseline. All reported scores use a beam size of 3 and Yu’s rerank mechanism for evaluation. Here are our takeaways:

- Our most successful experiment is mCE with normalized embeddings and partial softmax. It performs as well as Yu’s SOTA on METEOR and marginally better on CIDEr. We attribute the small jump in CIDEr to noise because the conditions we used for this experiment resulted in a similarity matrix that is very close to cross entropy and because other mCE experiments produced slightly lower scores than Yu’s SOTA.
- The shortcomings of the KNN Experiment are twofold:
 - Some tokens, like “man” and “woman” are assigned very high similarity scores, perhaps because embeddings are derived from the context of a word and “man” and “woman” have similar contexts in the corpora the embeddings were derived from. Therefore the KNN mCE matrix guides the algorithm to allow these tokens to be used in place of each other which is actually bad for the referring expression task, since a picture of a man should not be described as a picture of a woman.

- Some tokens, especially prepositions and positional words have very specific meanings. Consider the word “bottom”. In the training data, this word is used mostly to denote a positional attribute of an object, but its closest neighbor using cosine similarity is “behind” perhaps from its other meaning, meaning a person’s “bottom” or “behind”. Therefore the KNN mCE matrix guides the algorithm to allow these tokens to be used in place of each other which is actually bad for the referring expression task, since when used to denote a positional attribute of an object, these words have vastly different meanings.
- The handpicked synonym experiment serves as a controlled test to confirm our initial intuitions. However, this tests performs slightly worse than Yu’s SOTA.
- The synset synonyms tests performs substantially worse because many words in the vocabulary such as propositions do not have useful synonyms. Using antonyms to populate the similarity matrix clearly does not perform well.

6. Conclusion

Referring Expression Generation requires precise, discriminative language. While incorporating semantic information into an objective function may help a model generalize more effectively and prevent overfitting, sometimes that generality may actually decrease performance. We think that may be the case with our work. In other tasks, word pairs like man and woman, first and second, or baseball and basketball are all very similar. However, for the Referring Expression task, interchanging these words would be disastrous. Consequently, we think that semantically aware objective functions would be much better suited for other domains. We leave this to future work.

We also have concluded that word embeddings, when compared at the individual word-level as opposed to the sentence- or document-level, do not effectively encode semantic meaning. While this area is ripe for future exploration, initial results do not seem promising. However, supervising embeddings and contextual embeddings could ameliorate this issue.

Differentiability and positional encoding remain concerns for objective functions similar to ours. While the Gumbel-Softmax trick did not work with WMD, it has potential to work in other formulations, such as with WED. Hybridization and pretraining with our objective functions are also areas for future work; similarly, expected embeddings have shown promise in tandem with hybridization and pretraining and deserve future exploration.

Lastly, recent work has shown success integrating other non-differentiable objective functions via a Reinforcement Learning policy-gradient approach. We believe WMD and other semantically aware objective functions fit well with this approach.

References

- Bass, Jake, Jhaveri, Ishaan, and Norman, Bennett. Semantically aware objective functions for referring expression tasks. URL <http://bit.ly/2IynFTh>.
- Jang, Eric, Gu, Shixiang, and Poole, Ben. Categorical reparameterization with gumbel-softmax, 2016.
- Kusner, M. J., Sun, Y., Kolkin, N. I., and Weinberger, K. Q. From word embeddings to document distances. In *ICML*, 2015.
- wordnet.princeton.edu. About wordnet. URL <https://wordnet.princeton.edu/>.
- Yu, Licheng, Tan, Hao, Bansal, Mohit, and Berg, Tamara L. A joint speaker-listener-reinforcer model for referring expressions. *CoRR*, abs/1612.09542, 2016. URL <http://arxiv.org/abs/1612.09542>.