A Appendix

A.1 Details on Gromov Hausdorff

We briefly outline the procedure for computing the Bottleneck distance here. An interested reader can find further details at Edelsbrunner and Morozov (2013).

Computing the Gromov-Hausdorff distance involves solving hard combinatorial problems, but can be tractably approximated using the Bottleneck distance (Chazal et al., 2009). In order to compute the Bottleneck distance between two metric spaces, we compute the first order Vietoris-Rips complex (first order for computational efficiency) at t for both spaces: a graph containing an edge between two points iff they lie within a Euclidean distance t from each other in the metric space. As t is varied, the Vietoris-Rips complex goes from the individual points (at $t = 0$) to a single cluster (at $t = \infty$). As t increases, clusters are formed (birth) and eventually merge together (death). The persistence diagram is a 2D plot of the (t_{birth}, t_{death}) of each cluster, where t_{birth} and t_{death} are the values of t at which the cluster was born and died respectively. Given two persistence diagrams f, g, let γ be a bijective map from the points of f to the points of g. The bottleneck distance ($\mathcal{B}$) is then defined as:

$$\mathcal{B}(f, g) = \inf_{\gamma} \left(\sup_{u \in f} ||u - \gamma(u)||_{\infty} \right) \quad (7)$$

Chazal et al. (2009) showed that the Gromov-Hausdorff distance can be lower bounded by the Bottleneck Distance between the Persistence Diagrams of the Vietoris-Rips Filtration of the two spaces.

A.2 Analyzing Model Errors

We characterize the mistakes made by the model, and find that most fall into the following 4 categories:

Polysemy on the target side: These are the cases in which the predicted words and the gold translation are synonyms/hypernyms/hyponyms of each other.

Polysemy on the source side: These are the cases in which the gold translations and the predicted words are *different senses* of the source word.

Antonyms: The distribution of the context of antonyms is often very similar. Unsurprisingly the word vectors of antonyms are quite similar. This leads to cases where the predicted words and gold labels are antonyms of each other.

Words that occur in common contexts: Words that occur in numerous contexts often have poor word embeddings, since a single embedding can't capture polysemy. Consequently, multiple such word embeddings that are frequent and have poor representations often get incorrectly translated to each other. Some examples include proper nouns and numbers

We quantitatively estimate the fraction of errors due to these reasons using WordNet synsets. Given 2 synsets, WordNet provides a score denoting how similar two word senses are, based on the shortest path that connects the senses in the is-a (hypernym/hypnoym) taxonomy. The score is in the range 0 to 1. A score of 1 represents identity i.e. comparing a sense with itself will return 1.
We approximate the fraction of target polysemy errors by finding those cases for which the aforementioned similarity scores between the synsets of the predicted words and the gold translations ≥ 0.1. Similarly we approximate the fraction of source polysemy errors by finding those cases for which the similarity scores between the synsets of the source word and the predicted word ≥ 0.1. Fig 3 shows these estimations for different language pairs. See Table 6 for examples sampled from each of these error types.

A.3 β orthogonality projection vs. autoencoding loss

Lample et al. (2018) constrained the mapping matrix to be close to the manifold of orthogonal matrices by applying the following projection step after every update.

$$W \leftarrow (1 + \beta)W - \beta(WW^T)W$$

In our experiments we found out that the final accuracy is highly sensitive to the value of the hyperparameter β (Table 7). Our approach on the other hand uses an autoencoding loss which allows the model to flexibly adjusts the degree of orthogonality in a data driven manner and works consistently well for one choice of the scaling of the autoencoding loss.

A.4 Hyper-Parameters

The following are the hyper parameters used in the experiments. The values separated by / are the different values tried in the parameter search.

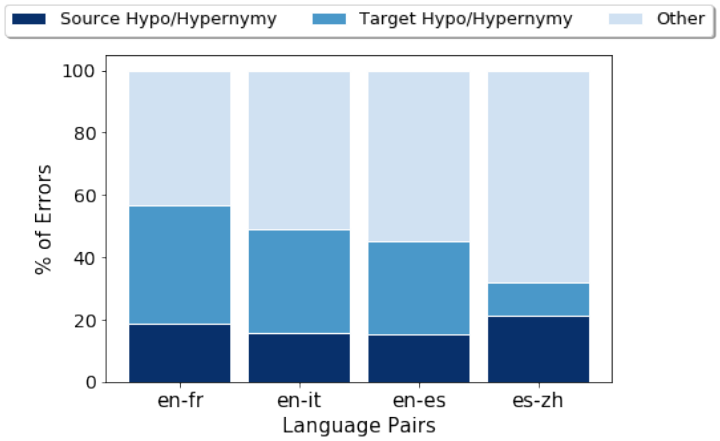

Figure 3: Fraction of errors coming from polysemy in the source/target side and antonymy, for the language pairs en-zh, en-it, en-es and en-fr

Type of Error	Source	Gold	Predicted	Comments
Target Polysemy	Shadows	影子	阴影	synonyms
Target Polysemy	Quest	Quest	Avventura	synonyms
Source Polysemy	Worn	usé	vêtement	Gold: used, Predicted: cloth
Source Polysemy	Bitter	苦	辛辣	Gold: bitter (taste), predicted: bitter (feeling)
Antonyms	Unofficial	Ufficiale	Funzionario	funzionario: official
Antonyms	Mature	Mature	Jeune	Jeune: young
Antonyms	Afraid	Paura	Contento	Gold: fear, Predicted: happy
Common Words	Everybody	Jeder	Spaß	Gold: Everybody, Predicted: Fun
Common Words	Fourteen	Vierzehn	Dreirzehn	Numbers translated incorrectly

Table 6: Sampled Errors

Lang	**Ortho**	β			**Auto**
		1e-2	1e-3	1e-4	
en-de	19.9	74.8	67.4	73.7	74.3
en-ru	102.5	40.8	30.7	36.7	46.1
en-zh	171.1	0	23.8	32.1	33.3

Table 7: Unsupervised accuracies for different values of β (MUSE) and our autoencoding loss.

- Number of words per language considered for GAN training: top 75000
- **Discriminator Parameters**:
  - embedding dim: 300
  - hidden layers: 2
  - hidden dim: 2048, 2048
  - dropout prob: 0.1 (Only on the input layer)
  - label smoothing: 0.1
  - non-linearity: LeakyReLU ($\alpha = 0.2$)
- **Generator Parameters**
  - Initialization: Identity / Random Orthogonal
  - Mean Center: True
- **GAN Training Parameters**
  - batch size: 32
  - Optimizer: SGD
  - Supervised loss optimizer: SGD / Adam
  - lr: 0.1 (with a schedule of 0.98 decay per round, and halved if unsupervised CSLS metric does not improve over two rounds).
  - Hubness Threshold: 20
- $f_a = cosine$