Characterizing a Latent Representation of Sentences

George Du, Jingyu Li, Brian Shimanuki, Vincent Tjeng

December 16, 2016

Abstract

Language allows for stylistic decisions by the writer even when the content is fixed. In this work, we introduce and study a variational autoencoder generative model that includes a continuous latent representation of English sentences separable in style and content. We present results demonstrating that the style dimensions are sufficient to predict authorship, and that style can be transferred effectively from one sentence to another.

1 Introduction

Languages allow for many syntactically correct sentences that express the same meaning. We say that each of these sentences differ in style, but not in content. Our goal is to find a compact, continuous latent representation of sentences that is separable in style and content, by explicitly designating dimensions in the latent representation to correspond to each of these two aspects of sentences.

We demonstrate that the portion of the latent representation designated to correspond to style contains the majority of the information required to distinguish the identity of the author for a sentence. We also demonstrate that by varying the style component of the latent representation, we can generate coherent sentences with the same meaning but with significantly different styles.

2 Related Work

An autoencoder consists of an encoder, which maps an element in the input space to a latent representation, and a decoder, which converts a latent representation back to an element in the original input space. The key difference between a variational autoencoder (VAE) and a regular autoencoder is that VAEs do not use a deterministic encoding function; rather they learn a posterior distribution, $q(\vec{z}|x)$, for the encoder. The prior distribution on z is constrained to be a unit Gaussian. As a result, VAEs learn a smoother and more interpretable feature space for sentence encoding, and latent representations that are generated by interpolating between representations for existing sentences can be decoded as coherent novel elements.

VAEs have had success in finding latent representations of objects ranging from images [5] to sentences [1]. While VAEs for sentences are able to explicitly model high-level properties (such as topic or overall syntactic features), existing architectures are not able to identify the dimensions of the latent space that correspond to each of these properties. Successfully segmenting the latent space would allow for many applications. One possible application is style transfer for sentences, where the content of a sentence is combined with the style of a different writer. This is similar to recent work in computer vision [3].

3 Experimental Approach

Our approach is largely based on the architecture presented in [5]. The crucial distinction is that, rather than training our VAE with single sentences - as would be typical in a standard autoencoder - we train the VAE with sets of sentences that match on either style or content. Prior to training, we explicitly designate a portion of the dimensions of the latent space as corresponding to style, with the remaining dimensions corresponding to content. The fraction of the latent space corresponding to style is an adjustable parameter, λ_s. Our results use $\lambda_s = 0.1$ In the case where a pair of sentences match on content but differ on style, we would expect that their latent representation varies only in the style dimensions, as shown in Figure 1 (The reverse is true for pairs of sentences matching on style but differing on content).

Optimizing a standard VAE involves optimizing

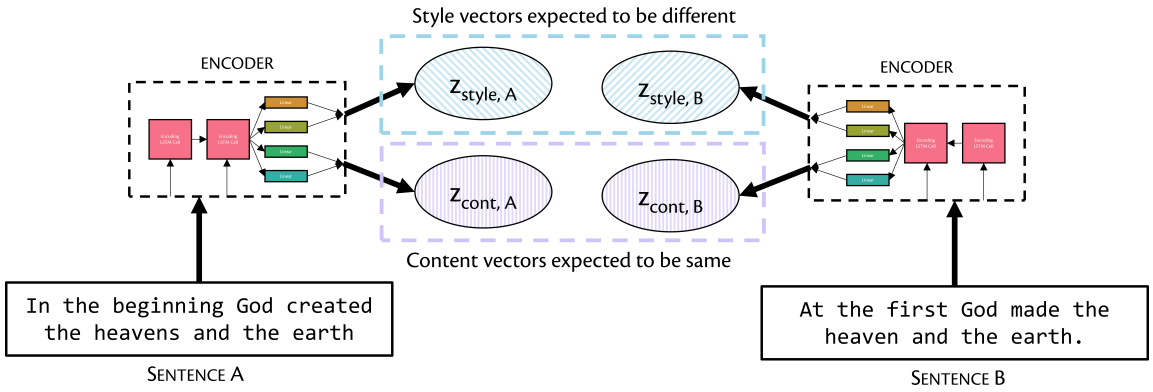


Figure 1: Visualization of paired training process for sentences matching in content but not in style.

a *variational lower-bound*[1], which is the difference between **1)** the expected log-likelihood of the original sentence conditioned on the recognition model, $\mathbb{E}_{q_\theta(\vec{z}|x)}(\log p_\theta(x|\vec{z}))$ and **2)** the KL-divergence $\mathrm{KL}(q_\theta(\vec{z}|x)\|p(\vec{z}))$. This ensures that sentences can be decoded with reasonable accuracy while maintaining the smoothness of our feature space.

To ensure that our the trained VAE separates style and content in the latent space, we add to our VAE a penalty term, shown in Equation 2, for unexpected differences (in the style or content dimensions, as appropriate) in the latent representation of pairs of sentences.

3.1 Corpus

A prerequisite for successful training of our VAE is *matching pairs*: that is, pairs of sentences that communicate essentially the same content but are written differently[2]. Our selected corpus of matching pairs consists of aligned sentences from multiple different translations of the Bible.

3.1.1 Corpus Details

The Bible is a compilation of many shorter books. These books were originally written in either Biblical Hebrew / Aramaic (for the Old Testament) or Koine Greek (for the New Testament) [8]. Multiple Modern English translations of the bible exist, and the total number of distinct translations, including translations of only parts of the bible, is estimated at 900 [2]. We used a subset of 27 of the most popular versions, which are listed in Table 5, rejecting translations that contain only a portion of the books in the King James Version and translations that used Hebrew names (such as the Orthodox Jewish Bible) or Old English (such as the Tyndale Bible). A key assumption here is that the variation in style between the authors of the *original* books is less than the variation in style between translations.

3.1.2 Sentence Alignment

To align sentences from different versions of the Bible, we take advantage of the fact that sentences in modern translations are indexed. Specifically, every book is divided into chapters; each chapter is in turn subdivided into verses spanning approximately one sentence [3]. The total number of verses is approximately 31,000. With the exception of approximately 20-30 verses in the New Testament [6], every translation of the Bible used has the same verses, with the content of the sentences being the same. Aligning to obtain matching pairs is thus straightforward.

3.1.3 Summary Statistics

Across all translations, the mean number of tokens in each verse was 29.5, the median number was 26.0, and the standard deviation of the number of tokens was 14.3. (The maximum number of tokens in a sentence is 406). The distribution of tokens in each verse is visualized in Figure 2. We operated on sentences of less than 30 words. The pruning is necessary to have reasonable LSTM performance, and the cutoff of 30 allowed us to retain over half of the training data.

[1] We call this term a variational lower-bound since it is a valid lower bound on the log-likelihood of the sentence x

[2] Differences in style could range from the placement of a prepositional phrase (at the beginning vs. at the end of a sentence) to how sophisticated the text is (casual vs. formal)

[3] A sentence may span more than one verse, and a verse may contain multiple sentences

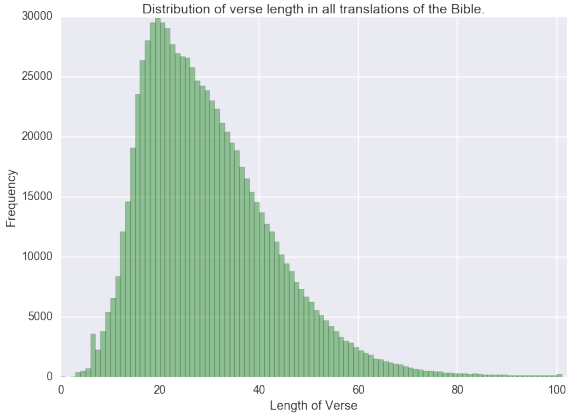


Figure 2: Distribution of number of tokens in verses across all translation of the Bible. The rightmost bin contains all verses greater than length 100.

3.2 Word Embedding

We used the `word2vec` model [7] to learn dense vector representations of the words in our corpus of size $d = 200$. A continuous bag of words model was used, which took each verse from every translation as an input. Each verse had an `EOS` token appended to it. Words that appeared less than 5 times in the entire corpus were labeled with the `UNK` token. This resulted in a `word2vec` model with vocabulary size 25,334.

3.3 Variational Autoencoder Architecture

The architecture for the variational autoencoder is shown in Figure 3.

Encoder At the high level, the encoder $q(\vec{z}|x)$ is a Gaussian $\mathcal{N}(z|\mu(x;\theta), \Sigma(x;\theta))$, where μ and Σ are deterministic functions with parameters θ. We constrain Σ to be a diagonal matrix. Specifically, the encoder consists of a single-layer LSTM [4] RNN, followed by fully connected linear layers producing means $(\mu_{style}, \mu_{cont})$ and variances $(\sigma_{style}, \sigma_{cont})$. The latent representation z_i is drawn from $\mathcal{N}(\mu_i, \sigma_i^2)$. The overall latent representation is $z = (z_{style}, z_{content})$.

Decoder The decoder consists of a single-layer LSTM RNN. We greedily decode sentences by selecting the word that has the largest dot product with the output of each LSTM cell, and feeding the word as input to the next cell. During training, the correct word is always fed into the LSTM.

The size of the LSTMs match the dimension of the word embedding, d, while the total number of dimensions in the latent space is $n = 2d$. z_{style} has n_{style} dimensions, while z_{cont} has n_{cont} dimensions; $n_{style} + n_{cont} = n$ and $n_{style} = \lambda_s \cdot n$.

3.4 Training and Loss Function

We start with a pre-training phase, where we minimize the negative of the variational lower bound for a single sentence without considering any segmentation-related penalty terms.

$$\mathcal{L}(x;\theta) = -\mathbb{E}_{q_\theta(\vec{z}|x)}(\log p_\theta(x|\vec{z})) + \mathrm{KL}(q_\theta(\vec{z}|x)\|p(\vec{z})) \quad (1)$$

Pre-training only operates on single sentences. The aim of this phase is to just train a conventional VAE to effectively reconstruct sentences. This phase lasts for 80 epochs.

After pre-training, we attempt to alter the learned VAE parameters, and separate style and content by segmenting the latent representation. To this end, we use sets of four sentences $X = \{x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}\}$, where $x_{i,j}$ indicates a sentence from translation i and verse j. By taking a Cartesian product over pairs of translations and verses, we can simultaneously compute and penalize style and content differences in the latent space. An example set is shown in Table 1.

We additionally define *pairwise latent penalties*, which are related to the difference in the latent representation of sentences pairs:

$$\mathcal{L}_z(a,b;\theta) = \begin{cases} \lambda_s \left\|z_{style,a} - z_{style,b}\right\|_2^2, & \text{if } a,b \text{ are from the same translation} \\ (1-\lambda_s)\left\|z_{cont,a} - z_{cont,b}\right\|_2^2 - \lambda_s \left\|z_{style,a} - z_{style,b}\right\|_1, & \text{if } a,b \text{ correspond to the same verse} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

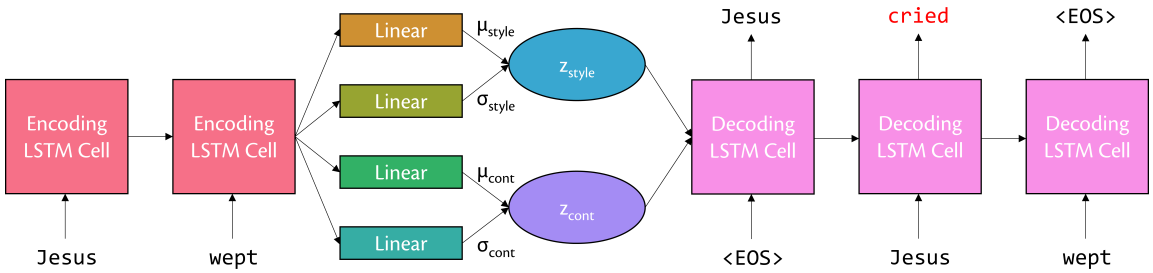


Figure 3: Architecture of our variational autoencoder. Words are represented as word vectors using the `word2vec` embedding.

	NIV	NKJV
1 Chronicles 16:11	Look to the LORD and his strength; seek his face always.	Seek the Lord and His strength; Seek His face evermore!
Colossians 3:2	Set your minds on things above, not on earthly things.	Set your mind on things above, not on things on the earth.

Table 1: Examples of paired sentences used in training.

When sentences a, b are from the same verse, we have a penalty on diverging on content $\|z_{style,a} - z_{style,b}\|_2^2$ as well as a reward for diverging on style $-\|z_{style,a} - z_{style,b}\|_1$. On the other hand, when sentences a, b are from the same translation, we only have the penalty term for diverging on style, since we expect some stylistic variation even for verses from the same translation.

The overall loss on our training set is the sum of the variational upper bounds for the four individual sentences, and the weighted pairwise latent penalties for each of the six pairs of sentences in X:

$$\mathcal{L}_{tot}(X;\theta) = \sum_{x \in X} \mathcal{L}(x;\theta) + w_{pen} \sum_{u,v \in X, u \neq v} \mathcal{L}_z(u,v;\theta) \quad (3)$$

w_{pen} is an adjustable parameter, and $w_{pen} = 0.5$ for our reported results. The training phase is 120 epochs.

Selecting w_{pen}. The weighting placed on the pairwise latent penalties in the overall objective was optimized as a hyperparameter. Figure 4 shows the impact of increasing w_{pen}, showing improvements in separation between the as w_{pen} increases.

3.4.1 Training Challenges

KL-divergence. As in [1], we found that straightforward training of the VAE caused the KL-divergence to be reduced to zero and the posterior distribution $q(\vec{z}|x)$ for all x to be identical to the Gaussian prior $p(\vec{z})$. Essentially, the neural net is trapped in a local minimum where it ignores the encoder and relies only on the decoder to reconstruct sentences. To address this problem, we apply the KL cost annealing used in [1], where the weight on the KL term is increased from 0 at the start of training to 1 at the end of training. This allows the model to encode as much information in $\vec{z}$ as possible during the beginning of the training process, but progressively forces the model to smooth out the posterior distribution towards the end of the training. We also experimented with weighting the KL-divergence by some constant factor w_{kl} in the loss function lower than 1. The form of the loss function then becomes

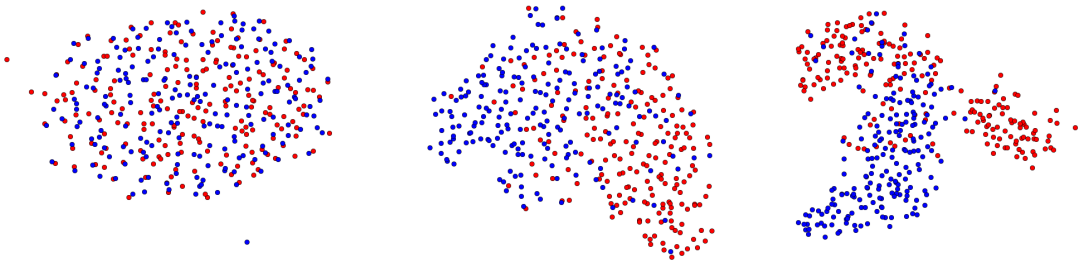
$$\mathcal{L}(x;\theta) = -\mathbb{E}_{q_\theta(\vec{z}|x)}(\log p_\theta(x|\vec{z})) + w_{kl}\mathrm{KL}(q_\theta(\vec{z}|x)\|p(\vec{z}))$$

This approach treats the KLD as a regularizer and sacrifices the ability to directly optimize a lower bound on likelihood. However, we found that it produced better performance while still retaining a well-structured latent space, and we report results using $w_{kl} = 0.2$.

4 Results

4.1 Style Classification

Style classification was carried out by extracting sentences from two different translations, training a binary classifier using a logit model on the feature vectors representing the sentence, and evaluating the



(a) $w_{pen} = 0$ (b) $w_{pen} = 0.25$ (c) $w_{pen} = 0.50$

Figure 4: t-SNE 2D embedding of style latent representation of verses. Red points are from the NIV translation while blue points are from the NKJV.

performance of the classifier on a test set. The baseline classifier and our VAE classifier differ only in the feature vector.

Baseline Sentences were represented as a bag of words, with the `UNK` token used for words not in the `word2vec` vocabulary. The dimension of each feature vector is 25,334.

VAE-Style The sample style latent representation z_{style} was taken from sentences encoded by the trained VAE. The dimension of each feature vector is 40.

VAE-Both The entire sample latent representation z was taken from sentences encoded by the trained VAE. The dimension of each feature vector is 400.

The VAE-Style classifier performs better than the baseline classifier in most pairs of translations we tested, and matches the performance of the VAE-Both classifier, demonstrating that the style latent representation contains most of the style information encoded in the latent space. Full results are shown in Table 2.

4.2 Exploring the Latent Space

4.2.1 Homotopies

The smooth nature of the latent space allows us to obtain coherent sentences from linear interpolations between sentences. The linear interpolation, or *homotopy*, between sentences a, b, is defined as the set of sentences formed by decoding points on the line $\vec{z_h}(t) = \vec{z_a}(1-t) + \vec{z_b}t$, where $t \in [0, 1]$. As shown in Table 3, the interpolated sentences are generally grammatical and form a smooth transition between the two different translations, indicating that our VAE is learning good representations in the latent space.

4.2.2 Style Transfer

We also wanted to look at style transfer, or the variation of sentences in the latent space with the same values over the content subspace but varying in the style subspace. We compared verses from the NKJV and NIV translations, using the content subvector from NKJV but interpolating the style subvector between the translations. For translations a, b, we decode from points on the line $\vec{z_s}(t) = (\vec{z}_{style,a}(1-t) + \vec{z}_{style,b}t, \vec{z}_{cont,a})$, where $t \in [0, 1]$. As seen in Table 4, the latent space has variations over the style subspace. Sometimes the style transfer does not form the original b sentence well, indicating that the actual styles and contents are not always completely separated into orthogonal style and content subspaces.

5 Conclusion

This paper demonstrates that it is possible to find latent representations of sentences via a variational autoencoder that are well segmented into style and content components. We find that the style component of the latent representation of a sentence contains most of the information needed to classify its source, and display some examples of homotopies and style transfer.

This was a challenging problem, and performance was not perfect. Manual inspection showed that many of our style transfer attempts yielded intermediate and final sentences that differed also in content. We believe that after training the neural net to separate style and content, the segmentation was not perfect, i.e. the designated style and content neurons represented a biased mixture of the two, and perhaps the neural network's interpretation of style diverges from a human's interpretation of style.

In future work, we could experiment with different encoder and decoder architectures. In particular,

Trans 1	Trans 2	Baseline Accuracy	VAE Style Accuracy	VAE Both Accuracy
NIV	NKJV	80.7%	85.6%	84.9%
NIV	ESV	74.3%	72.5%	73.0%
NIV	NIRV	90.2%	91.1%	90.7%

Table 2: Comparison of accuracy of different classifiers.

NKJV	**their idols are silver and gold , the work of men 's hands .**
0.0	*their idols are silver and gold , the work of men 's hands .*
0.2	*their idols are silver and gold ,*
0.4	*their idols are silver and gold , but the hands of men .*
0.6	*their idols are silver and gold , but gave it by the hands of human*
0.8	*their idols are silver and gold , but by selah .*
1.0	*but their idols are silver and gold , made by human hands .*
NIV	**but their idols are silver and gold , made by human hands .**
NKJV	**your righteousness is an everlasting righteousness , and your law is truth .**
0.0	*your righteousness is an everlasting righteousness , and your law is truth .*
0.2	*your righteousness is your righteousness , and an everlasting truth is law .*
0.4	*your righteousness is your righteousness , and is an everlasting truth .*
0.6	*your righteousness is your law everlasting is an everlasting righteousness and truth .*
0.8	*your righteousness is your law and everlasting is true .*
1.0	*your righteousness is true and your law is everlasting .*
NIV	**your righteousness is everlasting and your law is true .**

Table 3: Sample homotopies between sentences containing the same content but differing style. The original sentences are labeled with the translation they are found in, and the intermediate sentences are decoded from $\vec{z_h}(t)$ for the specified value of t.

NKJV	**then moses went up into the mountain , and a cloud covered the mountain .**
0.0	*then moses went up into the mountain , and a cloud covered the mountain .*
0.5	*then moses went up to the mountain , saying ,*
1.0	*then moses went up to the mountain , the cloud covered themselves .*
NIV	**when moses went up on the mountain , the cloud covered it ,**
NKJV	**the word of the lord came to me again , saying ,**
0.0	*the word of the lord came to me again , saying ,*
0.5	*the word of the lord came to me , saying ,*
1.0	*the word of the lord came to me ,*
NIV	**the word of the lord came to me :**
NKJV	**and he took the fortified cities of judah and came to jerusalem .**
0.5	*and he took the cities of judah and came to jerusalem .*
0.5	*and he took the cities of judah came to fortified jerusalem .*
1.0	*he took the city of fortified cities and rebuilt .*
NIV	**he captured the fortified cities of judah and came as far as jerusalem .**

Table 4: Sample style transfer between sentences containing the same content but differing style. Intermediate sentences are decoded from $\vec{z_s}(t)$ for the specified value of t.

multi-layer LSTMs could be used if care is taken to prevent overfitting. We also hope to use a broader range of corpora for our paired training approach. One type of corpora that is likely to be easy to work with is translations of famous works of literature, such as Shakespeare's plays, Homeric Epics (*The Odyssey* or *The Illiad*), or *Beowulf*. Each of these works of literature has multiple translations in English. Another possibility that eliminates the need to have aligned sentences is to add a (probabilistic) author classifier to the neural net architecture that has access to only the style latent representation, adding a penalty term proportional to the cross-entropy loss of classification. This would allow training to be carried out with single sentences.

6 Contributions

George Du Implemented VAE framework with batching, and later implemented pair training. Trained the network and used tensorboard to visualize learning and tune parameters. Generated ideas for improving performance. Contributed to writing and editing the final paper, and created t-SNE visualization. Provided my NVIDIA GTX 1070 for training.

Jingyu Li Worked with George to implement the initial VAE architecture. Wrote code that allows us to view the output sentences. Implemented KL annealing. Implemented baseline classifier and classifier using our latent representation as features.

Brian Shimanuki Sanitized and tokenized training corpus. Constructed word vectors. Implemented generative RNN decoder that uses the previous output word for the next word. Handled batching for quadruples of sentences. Made interpolation and style transfer for a trained model.

Vincent Tjeng Explored appropriate corpora and obtained Bible training corpus. Refactored code to enable paired training approach. Debugged baseline classifier. Designed most of poster and contributed significantly to paper writing and contruction of visuals.

A Bible Translations

We used the 27 translations of the Bible listed in Table 5. Each of these translations contained all the books of the Bible found in the King James Version (KJV), and do not contain the Apocrypha.

ID	Name
ASV	American Standard Version
BBE	The Bible in Basic English
CEB	Common English Bible
CJB	The Complete Jewish Bible
CSB	Holman Christian Standard Bible
DBY	The Darby Translation
ESV	English Standard Version
GNT	Good News Translation
GW	God's Word Translation
JUB	Jubilee Bible 2000
KJV	King James Version
LEB	Lexham English Bible
MSG	The Message Bible
NAS	New American Standard Bible
NCV	New Century Version
NIRV	New International Reader's Version
NIV	New International Version
NKJV	New King James Version
NLT	New Living Translation
NRS	New Revised Standard
RHE	Douay-Rheims Catholic Bible
RSV	Revised Standard Version
TMB	Third Millenium Bible
WBT	The Webster Bible
WEB	World English Bible
WYC	Wycliffe
YLT	Young's Literal Translation

Table 5: Translations of the Bible used.

B Code

Our code is located at
`https://github.com/bshimanuki/6.864.git`

References

[1] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

[2] W.J. Chamberlin. *Catalogue of English Bible Translations: A Classified Bibliography of Versions and Editions Including Books, Parts, and Old and New Testament Apocrypha and Apocryphal Books.* Bibliographies and Indexes in Gerontology. Greenwood Press, 1991.

[3] Michael Elad and Peyman Milanfar. Style-transfer via texture-synthesis. *arXiv preprint arXiv:1609.03057*, 2016.

[4] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[5] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.

[6] Bruce Manning Metzger et al. *A textual commentary on the Greek New Testament*, volume 2007. United Bible Societies New York, 1971.

[7] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[8] Herbert Weir Smyth and Gordon M Messing. *Greek Grammar*. Harvard University Press, 1956.