Unsupervised Cognate Identification with Variational Autoencoders

Marlon Betz

August 13, 2016

Contents

1 Introduction

2 Motivation

2.1 Sound Change as a Walk in Feature Space

Sound change is usually described as a change of distinctive phonological features. Accordingly, a sound change such as final devoicing as in

$$MHG/hund/ \rightarrow NHG/hunt/ \tag{1}$$

would be captured by a rule like

$$[+\text{VOICE}] \rightarrow [\text{-VOICE}]/\_\# \tag{2}$$

which describes the loss of voice at the end of a word.

If we use

If we have established such a latent feature space $\mathbb{R}^n$, a sound change sc that derives a recent form of word w_{recent} from an ancient form $w_{ancient}$ should then correspond to a vector v_{sc} in such a way that

$$v_{w_{ancient}} + v_{sc} = v_{w_{recent}} \tag{3}$$

From this follows that

$$v_{sc} = v_{w_{recent}} - v_{w_{ancient}} \tag{4}$$

That is, we can formulate sound changes such as 2 without neither the actual sound change nor the conditioning specifying where the sound change should apply, but only the respective words involved. If we further assume that that sound change affects another word w', we have

$$v_{w_{recent}} - v_{w_{ancient}} = v_{w'_{recent}} - v_{w'_{ancient}} \tag{5}$$

which is equivalent to

$$v_{w_{recent}} = v_{w_{ancient}} + (v_{w'_{recent}} - v_{w'_{ancient}}) \tag{6}$$

If we want to evaluate that latent feature space, we investigate in how far that compositional structure is preserved in our latent space. In fact, such analogy tasks can be used as an evaluation method to test whether the learned embedding space encodes the structure expected to be inherently contained in the data [Mikolov et al.2013].

As we usually do not know the ancient version of a word but only recent ones, we expect the cognates to have spread in the feature space from the origin to some directions.

$$v_{w_{recent}} \sim \mathcal{N}(\mu = v_{w_{ancient}}, \sigma^2 I) \tag{7}$$

3 Related Research

4 Architecture

Hence, the model should have three major components:

1. The phonemes should be embedded in a feature space, where similar phonemes should cluster in similar subspaces of the feature space.

2. The words as sequences of such phoneme embeddings should themselves be embedded in another feature space, where words with similar shape should cluster among each other.

3. The word embeddings are then clustered in such a way that words that appear together in a cluster are assigned a common label, which is then predicted cognate class.

4.1 Phoneme Vectorization

4.1.1 Hand-crafted Vectorization Models

4.1.2 Data-driven Embeddings

4.2 Word Embeddings

4.2.1 Autoencoders

4.2.2 Variational Autoencoders

4.3 Clustering

4.3.1 Affinity Propagation

5 Evaluation

5.1 Data

5.2 Results

6 Resume

7 Acknowledgements

For training the phoneme embeddings, I used the word2vec implementations provided by the gensim package [Řehůřek and Sojka2010]. The Autoencoder was implemented with Keras [Chollet2015] and Tensorflow [Abadi et al.2015]. The clustering algorithms used here were provided by scikit-learn [Pedregosa et al.2011]. All code connected to this thesis can be found on my github [1]

References

[Abadi et al.2015] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[Chollet2015] Chollet, F. (2015). Keras. https://github.com/fchollet/keras.

[Mikolov et al.2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

[1] https://github.com/marlonbetz/BA

[Pedregosa et al.2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Řehůřek and Sojka2010] Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. http://is.muni.cz/publication/884893/en.