Visualizing Low-Dimensional Word Embeddings with Emoji Annotations

Yoshinari Fujinuma*
Department of Computer Science
University of Colorado Boulder

Shantanu Karnwal†
Department of Computer Science
University of Colorado Boulder

ABSTRACT

Word embeddings are quite a buzz in today's time in the lingusitics community, mainly because of their excellent performance in NLP applications like machine translation, topic modelling, question answering, etc. But when we try to look at the visualizations of these embeddings, we don't seem to get any sort of take-home knowledge from that. We propose 2D visualizations of these embeddings by using a clustering algorithm, and to have these clusters express their semantic information in a more understandable way, we annotate these clusters with emojis.

1 INTRODUCTION

Word embeddings [11] has been successful in multiple NLP applications such as machine translation [2, 9] and cross-lingual document classification [8]. The core idea of word embeddings is that assuming a distributional hypothesis [7] i.e., words that have similar context has similar meanings, it maps each word into a vector with dimensions typically ranging from 50 to 300. But one area where there has not been very specific research is directly extracting useful information from the embeddings themselves.

One way to efficiently convey this semantic information about embeddings is to enable an interaction between humans and the visualizations of these embeddings. However, there are problems of visualizing embeddings such as

- Overlap of words when there are lots of data points (Figure 1).
- It takes time for users to understand and analyze embedding space solely from words and its gemometric arrangements.

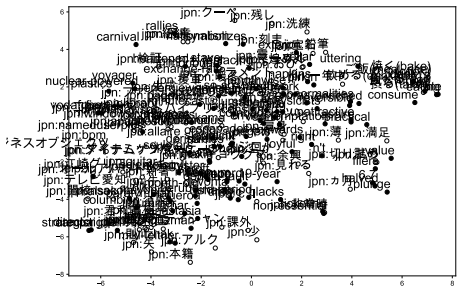

Figure 1: An example of a visualizaiton of word embeddings using t-SNE. Visualization of around 200 words causes clutter and makes humans hard to extract useful information.

The ultimate goal of any good visualization is to convey the user about everything the data represents, and not what the data is like. Therefore, we think that semantics is a crucial aspect of any good visualization. So when we decided to carry on this project, the key question we asked ourselves was - How can we represent word embeddings interactively in a 2D design space by keeping the clutter on the design space as minimum as possible?

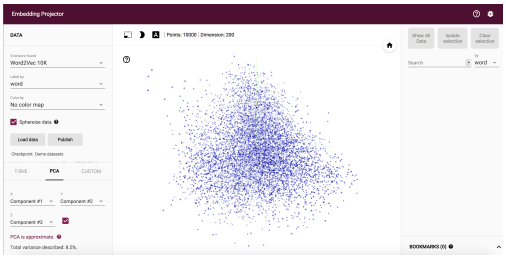

Figure 2: The visualizaiton of word embeddings using Tensorboard.

Therefore using this motivation, we carry forward this project, and accomplished the following:

- Used a 2D design space to visualize the entire word2vec embedding space.
- Kept the clutter minimized by having an efficient k-means clustering algorithm implemented on the cosine similarities of these word vectors.
- Convey the semantic information of every single cluster by annotating them with Emojis, because of their excellent way of conveying semantic information with just a single character.

2 RELATED WORK

As we know word embeddings, like in word2vec [12], have brought a drastic change in the lingusitics community. It is mainly because every early approach to NLP required a knowledge of linguistics and it was assumed that a person knowing more languages would intuitively know more about doing efficient natural language processing. But word embeddings came and just changed everything. Not only we need to have any sort of lingusitic knowledge to perform NLP applications, but we can just represent any word of any language as a very high dimensional vector. Since these embeddings are of a very high dimension, visualizing these embeddings becomes hard since we cannot visualize anything using more than 3 dimensions.

Fortunately, there have been many ways of visualizing word embeddings, mainly by using the principles of dimensitionality reuduction and projecting these embeddings onto a 2 Dimensional space. Two very prominent techniques to do that are Principal Component Analysis (PCA) [14] and the more recent t-SNE [13]. Both of these techniques bring in potential problems that make it for the user hard to understand anything from these visualizations. For example, when the number of data points become extremely large, the visualization gets largely cluttered. The visualization of word embeddings using t-SNE is shown in Figure 1. Not only that, but t-SNE is also known to have counter-intuitive features such as "cluster sizes mean nothing"[1]). One possible solution to avoid the visual cluttering is to enable user interaction and move around the

*e-mail: Yoshinari.Fujinuma@colorado.edu
†e-mail: Shantanu.Karnwal@colorado.edu

[1]https://distill.pub/2016/misread-tsne/

words.

One notable work on adding interaction on top of dimensionality reduction methods to avoid the visualization cluttering is Tensorboard (Figure 2) [1]. Tensorboard has multiple user-friendly features to enable exploratory analysis on word embeddings by (1) searching for words, and (2) clickabble and draggable 3D space. However, there are many downsides of a Tensorboard-based visualization. One is the lack of summarization and collapsing data points, and also that the user does not have any idea of how to make sense of this kind of a data, because all the user can see is a large collection of points distributed in the 3D space. So when we started this project, we understood that projecting embeddings on a low-dimensional space and just adding basic interaction will not help the user understand the semantics of the word embedding visualization.

Therefore in our project, we focus on adding an efficient data collapsing feature, achieved by an efficient k-means clustering algorithm, and also adding an appropriate feature to make the user understand the semantics of these word embeddings, which we achieve by using emojis.

3 Description of the Project

Our goal for this project is to visualize word embeddings by avoiding the clutters caused by too many data points. Our core idea for the approach is to collapse the data points into clusters and annotate with emoji to summarize the cluster. Figure 3 shows the example output of our visualization.

Figure 3: The visualization of word embeddings using clustered network and emojis.

Our approach for constructing the visualization is as follows:

1. Train a word embedding using a raw corpus.
2. Run k-means [10] on the word embedding space and obtain clusters
3. Assign an emoji to each cluster
4. Visualize using D3.js

For the number of clusters k, we empirically set $k = \{40, 50\}$. We only show the top 10 nearest neighboring words to the centroid to avoid the visualization clutter. This issue is further discussed in Section 4.

Figure 4: An example of expanding clusters after clicking on emojis.

3.1 Annotation of Clusters with Emojis

Searching for a right word to represent the cluster requires external linguistic resources e.g., WordNet and a sophisticated method e.g., hypernym prediction. However, humans are good at associating multiple words and capture the abstract meaning form a picture. For example, when one looks at the tomato emoji (🍅), the possible association of this words are "tomato", "vegetable", "food", or even "object". Therefore, we decide to use emojis to represent the clusters.

Another motivation of using emojis instead of words is the "Picture superiority effect" (e.g., [3]) i.e., humans are better remembering pictures than words. Assuming that the annotation of clusters is the most crucial component to summarize word embeddings, we use emojis to represent the clusters.

To assign an optimal emoji e_c to each cluster c with centroid vector v_c, assume we have a set of emoji and its text description t. We compute the mean word vector v_t of each word w_i in a given emoji description $t = \{w_1, w_2, ..., w_n\}$ with length n out of all set of emoji descriptions T, i.e.,

$$e_c = \text{argmax}_{t \in T} \cos\_\text{sim}(v_c, v_t) \quad (1)$$

where

$$v_t = \frac{\sum_{i=1}^{n} w_i}{n}. \quad (2)$$

Word vectors are trained by Skip-gram with negative sampling [11] using 1 million sentences of English news articles from Leipzig corpora collection [6]. We set the word vector dimension as 100. Emojis are obtained from dataset built by [4].

3.2 Interaction

To let the users dive deeper into further details of clusters, users can click on emojis to "drill-down" [5] the cluster and look into the nearest neighboring words in terms of the centroid in the cluster. However, even when we collapse words into each cluster, there are still overlapping words as shown in Figure 4. To solve this issue, we further made each cluster draggable and used forced layout to let users resolve the overlapping words.

4 Discussion

Emojis Captures Approximate Meaning One of the successful cases of our visualization is a cluster with the spacecraft emoji 🚀 annotated. This cluster contains transportation-related words such as "c-130" and "refueling".

On the other hand, some emojis (e.g., Leo emoji) are harder to interpret what does a cluster mean. Furthermore, one emoji could have multiple descriptions (e.g., the descriptions for the leo emoji are "greek", "sign", "zodiac", "stars", "constellation", "astrology", "lion") which further requires sophisticated processing when we want to improve the quality of the visualization.

Emojis are diverse Sometimes it just amazes us with how many emojis have been introduced over the years. We are not just talking about skin color emojis, but also things like country flags, food items, plants, animals, etc. This number is still keeping on increasing. So when we have so many emojis trying to represent the same kind of thing, what emoji should be used to annotate the cluster of the words inside it. For example, if all the country names are clustered together, what country flag emoji do we use to annotate it? Another example is about places in Europe, so we need something representing Europe to annotate that particular cluster. One emoji that we can think of is the European castle emoji. But currently the algorithm tries to annotate it with the Japanese castle emoji. This is a pretty interesting topic to think about, but for now we are leaving it for future work.

Outliers are painful, but can't be ignored Outliers occur in almost every data visualization task, so there is a chance of a cluster accomodating the outlier. So as a result, the emoji that captures the semantics of the cluster, cannot capture the outlier because they might be totally unrelated. But on the other hand, we cannot also ignore them because we still need to represent the data in its entirety. But since this is a very active debate, we leave the topic of visualization of outlier word embeddings as a future work.

We cannot avoid clutters when data points become large ($>$ 5000) The problem of visual cluttering is still unresolved when the number of data points we want to visualize become large. Since we only made the drill available by one level,even assigning 100 words per cluster would cause visualization clutter.

5 Conclusion

In this project, we used the combination of clustering and annotating those with emojis to relax the problem of visual cluttering and summarizing the visualization. The future work is to further use hierarchical clustering to enable multiple levels of drill down to resolve visual clutters when the number of data points become larger.

References

[1] Tensorboard: Embedding visualization — tensorflow. `https://www.tensorflow.org/versions/r1.1/get_started/embedding_viz`, 2017. Accessed: 2018-05-02.

[2] M. Artetxe, G. Labaka, E. Agirre, and K. Cho. Unsupervised neural machine translation. In *Proceedings of the International Conference on Learning Representations*, 2018.

[3] T. Curran and J. Doyle. Picture superiority doubly dissociates the erp correlates of recollection and familiarity. *Journal of Cognitive Neuroscience*, 23(5):1247–1262, 2011.

[4] B. Eisner, T. Rocktäschel, I. Augenstein, M. Bosnjak, and S. Riedel. emoji2vec: Learning emoji representations from their description. In *Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media*, 2016.

[5] N. Elmqvist and J.-D. Fekete. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *IEEE Transactions on Visualization and Computer Graphics*, 16(3):439–454, May 2010. doi: 10.1109/TVCG.2009.84

[6] D. Goldhahn, T. Eckart, and U. Quasthoff. Building large monolingual dictionaries at the Leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Language Resources and Evaluation Conference*, 2012.

[7] Z. Harris. Distributional structure. *Word*, 10(23):146–162, 1954.

[8] A. Klementiev, I. Titov, and B. Bhattarai. Inducing crosslingual distributed representations of words. In *Proceedings of International Conference on Computational Linguistics*, 2012.

[9] G. Lample, A. Conneau, L. Denoyer, and M. Ranzato. Unsupervised machine translation using monolingual corpora only. In *Proceedings of the International Conference on Learning Representations*, 2018.

[10] S. Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of Advances in Neural Information Processing Systems*, 2013.

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

[13] L. Van der Maaten and G. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 2008.

[14] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.