Word Vectors Project

Markus Gabriel
01326657

June 2018

The full source code for the project can be found at
`https://github.com/MGabr/word-vec-proj`.

1 Word Vector Training

To train word vectors I decided to use the word2vec approach and to use the latest dump of the german wikipedia available at `https://dumps.wikimedia.org/dewiki/latest/dewiki-latest-pages-articles.xml.bz2`. I downloaded the dump on May 19, so the current version at this URL will be different. To get an iterator over all articles of the dump, containing all words per article, I used `gensim.corpora.WikiCorpus` with `dictionary={None: None}` to avoid additional, not required processing. Using these iterators I built two word2vec models using `gensim.models.Word2Vec`.

As word2vec parameters I just used the default ones and only varied the window size. I trained one model with a default window size of 5 and another model with a small window size of 2.

2 Country Capital Relationship Visualization

To first experiment with visualizations and verify that my approach works, I tried to reproduce a common visualization like in [1] where one visualizes countries and their capitals with the results that the countries and their capitals are on different sides with similar vectors between each country and its capital.
To do this I chose an array of countries and their capitals, got the word vectors for all those words, performed PCA to two dimensions for those word vectors and then plotted the 2D vectors resulting from the PCA with lines connecting each country and capital. Figure 1 shows the resulting visualization when using the model created with a normal window size and figure 2 shows the visualization created when using the model with a small window size. One can observe that the model with the normal window size produces results as expected with lines being parallel to some degree while the model with the small window size achieves the separation of countries from capitals (left to right) but does not lead to parallel lines, with the line from finnland to helsinki even crossing two other lines.

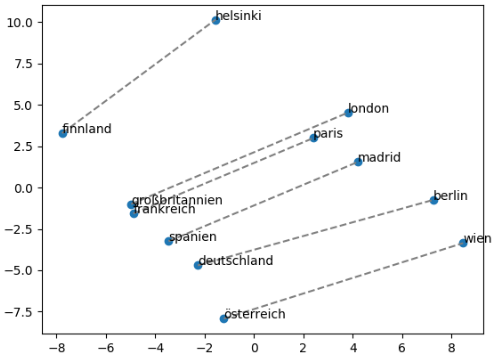

Figure 1: country and capital relations for window size 5

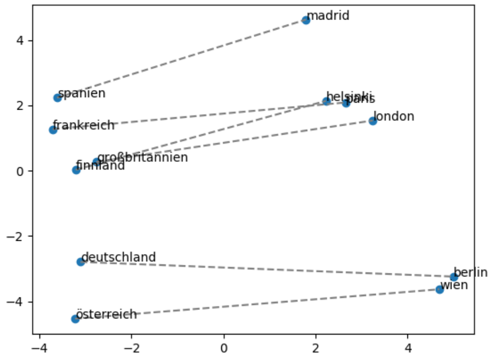

Figure 2: country and capital relations for window size 2

I also made an experiment trying to find the country for a capital. I therefore used the `most_similar` function from `gensim.models.keyedvectors` which finds the most similar words to a mean of positive and negative words. I used österreich as positive word and wien as negative word and then added each capital city once as additional positive word. So this basically built the mean of capital + österreich - wien. The predictions for the capitals were always correct, also for the model trained with the small window size.

3 Topic Relationship Visualization

As two topics to compare words from, I picked economics and natural science (physics and biology). As words with different meanings in those topics I picked the following.

- Gehalt (Einkommen vs Anteil)
- Länge (Ort vs Zeit)
- Leistung (Arbeitsleistung vs physikalische Leistung)
- Materie (Thema vs Materie in der Physik)
- Grenze (zwischen Ländern vs für Werte)
- Organ (Institut vs Organ einer Person)
- Arme (arme Personen vs Arme einer Person)
- Arm/arm
- erben (Geld erben vs Gene erben)

I got the word vectors for all words to visualize, performed PCA to two dimensions for those word vectors and then plotted the PCA vectors. All natural science vectors are plotted either in red (physics) or yellow (biology), all economics vectors are plotted in blue and the vectors of shared words with different meanings are plotted in gray.

One can see that for both models words from natural science are left and words from economics are on the right side of the figures. In addition, specific words like bakterien,

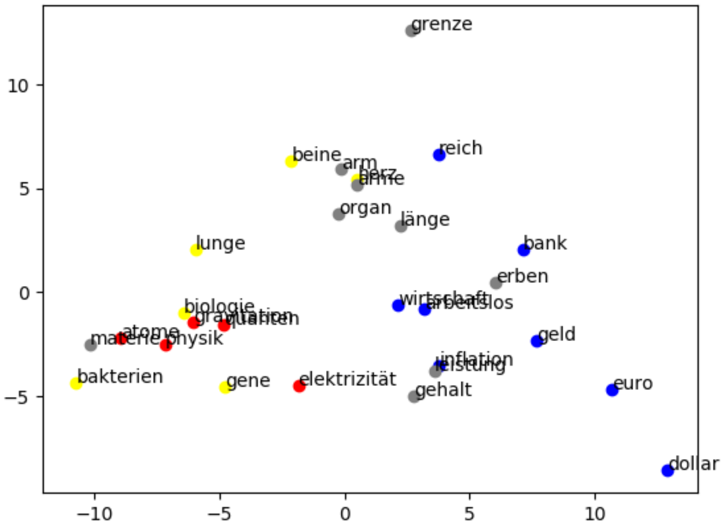

Figure 3: economy and natural science relations for window size 5

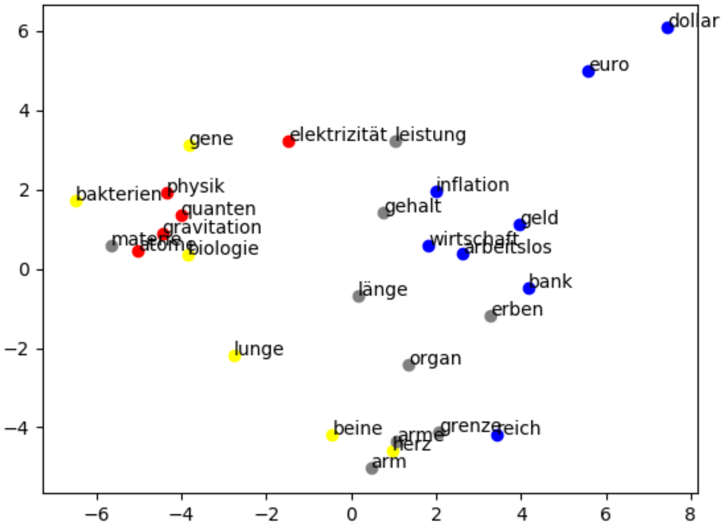

Figure 4: economy and natural science relations for window size 2

gene, euro and inflation are located rather at the bottom while more general words like grenze, arm, reich, herz and bank are rather at the top - or the other way around in the case of the model trained with a small window size. The very specific words I picked from physics are clustered all in a small area.

Most shared words are located somewhere between the topics which is a behavior I expected. They do, however, lean more or less towards the topic they are supposedly more important to. Notable observations are the distance from arm to beine and reich as well as erben to geld and gene. Besides just looking at the distances in the figure I also computed the similarity between the original word vectors. arm is significantly closer to bein (0.6806 for normal window size, 0.6407 for small window size) than to reich (0.1097, 0.2518) as well as to besitzlos (0.1448, 0.1754), which is not an opposite like reich. erben is in the middle of the economics words and therefore it is also not surprising that it is significantly closer to geld (0.3742, 0.4106) than to gene (0.0724, 0.1099). When using vererben instead of erben the distance is reduced a bit, but is still significant. While the fact that arm is closer to bein than to reich might be surprising, erben being closer to geld than gene is not suprising at all and also fits the amount of google results returned for the word combinations.

Apart from the different orientation, the model trained with a small window size does not differ that much from the model trained with a normal window size. The small window model even seems to separate the topics better since the common words are more in the middle and not as intertwined with the economics words. One notable observation here is that leistung is closer to elektrizität than to wirtschaft and also not right besides inflation like it is in the normal window model. Further organ is further from herz than in the normal window model and grenze is much closer to other words and not very separate from all other words. I can, however, not see an intuitive explanation for this behavior due to a lower window size. In general I would argue that the difference in window size between 5 and 2 was probably too small to have a significant impact on the created word vectors.

window	**positive**	**negative**	**top 5 most similar words**
5	leber, herz, lunge		niere, nieren, magen, milz, hirn
2	leber, herz, lunge		niere, nieren, gebärmutter, milz, schilddrüse
5	arm, bein		unterarm, handgelenk, unterschenkel, oberschenkel, ellbogen
2	arm, bein		unterarm, handgelenk, oberarm, oberschenkel, knöchel
5	lebendig, arm	reich	leblos, versteinert, verkrüppelt, fürchterlich, geistesabwesend
2	lebendig, arm	reich	leblos, verkrüppelt, gefesselt, sehend, oberkörper
5	euro, dollar, yen		gbp, rupien, peseten, chf, eur
2	euro, dollar, yen		peseten, gbp, rupien, rubel, pesos

Table 1: Top 5 most similar words to averages of different words for different window sizes

To further prove my point that both window sizes produce quite reasonable results, I performed experiments to get the most similar word to an average of some words. Table 1 shows the result of those experiments. In general the resulting 5 most similar words are quite similar and I would not say that one window size performs better than the other.

One interesting example is arm as positive word and reich as negative word. Since these are opposites, I hoped to predict the opposite for the other positive word lebendig, which I also did somehow when leblos was predicted. We can, however, see the influence the meaning of arm as bodypart seems to have since tot was not predicted and the terms leblos, verkrüppelt, versteinert and gefesselt are related rather closely to arm. e.g. ein lebloser Arm, ein verkrüppelter Arm, sein Arm war wie versteinert, seine Arme waren gefesselt.

4 Conclusion

To sum up, I did find some differences between word vectors trained with a normal window size and those trained with a small window size, but not really large differences. For tasks like finding similar words to the average of some words or to order words by topic in visualizations, both window sizes seem to perform well. Both models also performed well predicting the country for a capital. When, however, using the word vectors trained with a small word window size for visualizing the country and capital relation, they did not perform that well, which leads me to believe that, in contrast to the general tasks in section 3, for more detailed and specific tasks the window size does matter.

References

[1] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed Representations of Words and Phrases and Their Compositionality *In Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, 2013, Pages 3111-3119*