Deep Learning

Word Embeddings

Bon Jovi Lyrics

Olga Valls Murcia

Introduction

The aim of this Laboratory is to experiment with word embeddings and find semantic similarities between words. Instead of using pre-trained word vectors, a vocabulary has been created using Gensim implementation for Word2Vec.

Dataset

The data set constructed for this work consists on 176 lyrics from 15 Bon Jovi albums, creating 7890 short sentences, that can be found in the following url: https://www.azlyrics.com/b/bonjovi.html

What do we want to do?

As it's known, content is closely related to meaning. With word embeddings, we convert words into vectors as a list of numbers that describe the word. That number is the number of neurons in the hidden layer of the feed-forward Neural Network.

In this work we test different sizes for the vectors that describe the word and different lengths of window, which is the context in which each word exists. This gives us different 2-dimensional space representations for words as well as for lemmas.

Lemmas representation is smaller than word representation; the vocabulary decreases as only nouns, verbs, adjectives and adverbs are taken from the original word set. That makes the spatial representation easier to read.

Tests

Testing 29 lyrics

A first test with 29 lyrics has been done in order to see which results are given with few words/lemmas in the vocabulary. As the parameters change, so do the plots, for words as well as for lemmas. In order to generate the plots, the word vectors are compressed in 2D space.

The Word2Vec parameters modified are the following:

- size: size of the vector that describes each word.
- min_count: the words that appear less than min_count are deleted from the dictionary (the dictionary is pruned).
- window: how many words in the context are taken into account to represent a word.

```
model = gensim.models.Word2Vec(sentences, size=150, window=10,
min_count=2, workers=10)
```

As all the figures show, the representation for words (left, in blue) and lemmas (right, in red) are totally different. Lemmas only have nouns, verbs, adjectives and adverbs in his neighbourhood, and words can have any kind of word; it could be the case that a noun had as close neighbour a determiner as it usually appears in the lyrics close to it.

In Figures 1 and 2 below, the vector size is 150, the window is 10 words/lemmas, and the parameter min_size is the one that has been tested. As the number increases, less words and lemmas are shown, as a stricter pruning in the vocabulary has been made.

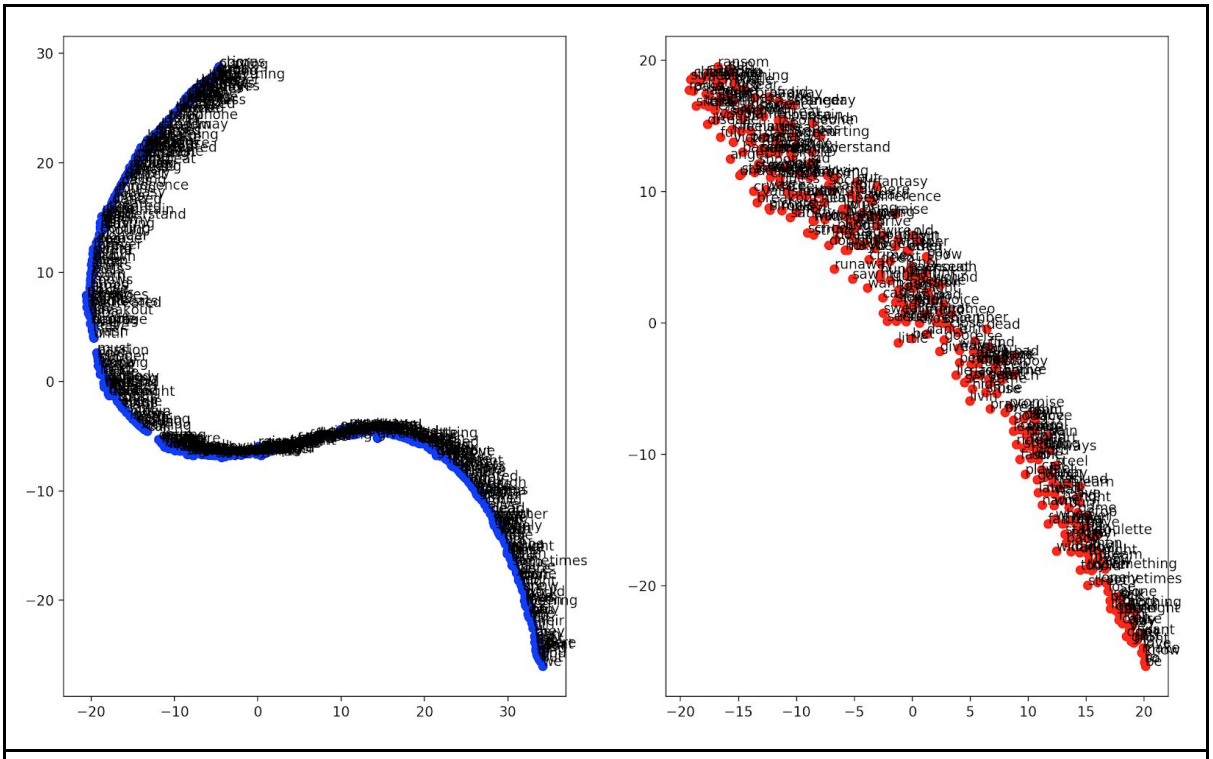

Figure 1: min_size = 2 (words left, lemmas right)

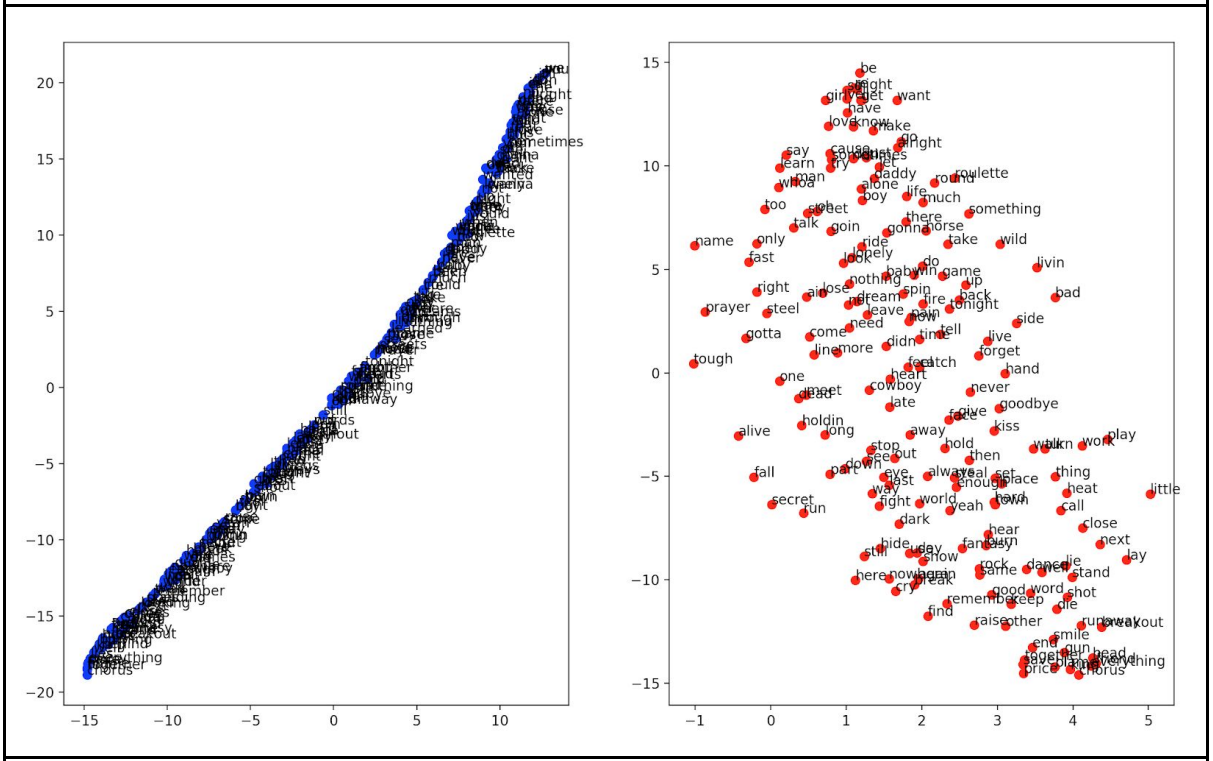

Figure 2: min_size = 5 (words left, lemmas right)

A min_size of 2 produces a lot of words/lemmas and, given the number of lyrics of the dataset, the lower complexity of songs (compared to sentences from a book), and the repetition of sentences that the lyrics tend to have, it has been decided to fix the min_size to 5 for the rest of this work.

Figure 3 below shows the result for a vector size of 300 (which is the size usually used in NLP), a window of 10 words/lemmas, and min_size of 5. More neurons in the hidden layer help get more precision in creating the vectors given the context, which define the words/lemmas. In the previous example, with half the neurons, the lemmas were scattered in the space.

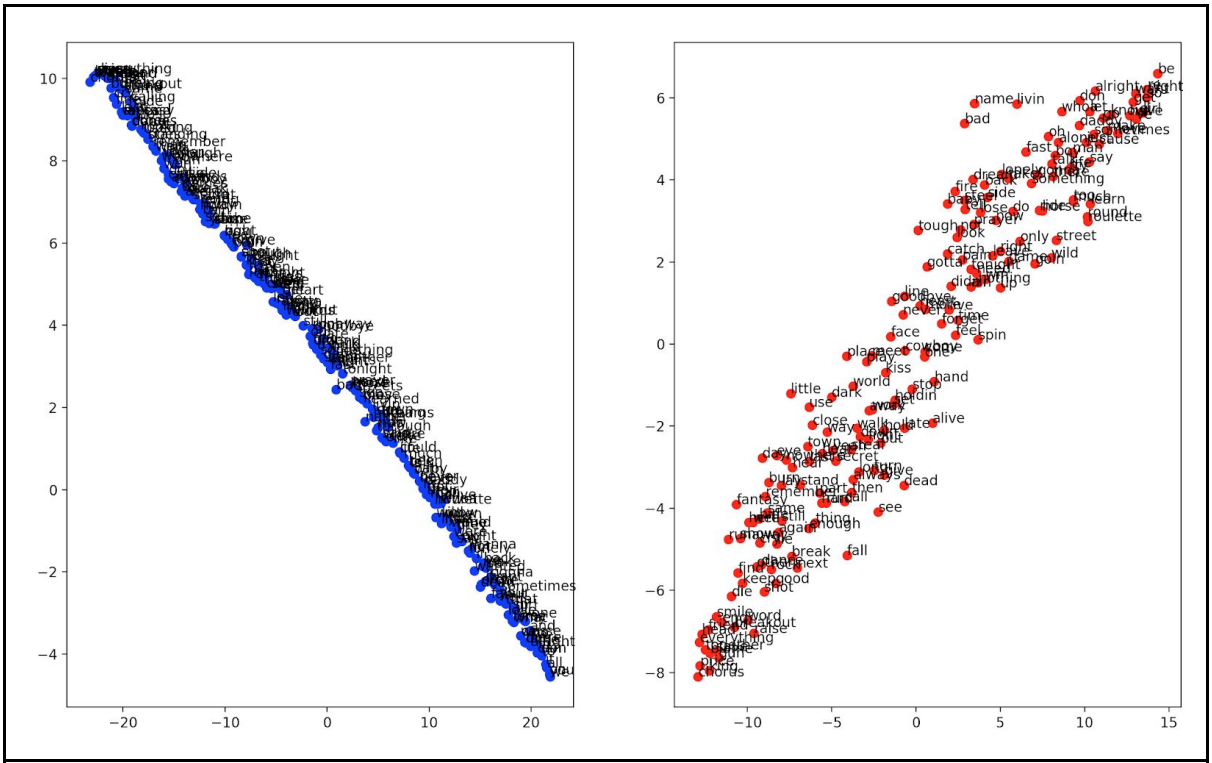

Figure 3: words left, lemmas right

Changing the window size

The window defines the context. The larger the context, the better its description would be. In the case of the sentences composing the lyrics of the songs, as they are very short, the window doesn't need to be very large, so for this work, a window of 5 would be enough.

And example for some sentences composing a piece of a lyric is as follows:

...
Goodnight New York goodnight
Don't ever close your eyes
If these streets could talk, New York, New York
So fine they named you twice
Your face, your brains
No other place
...

Figure 4 below shows the result of applying a vector size of 300, a min_size of 5 and a window of 5.

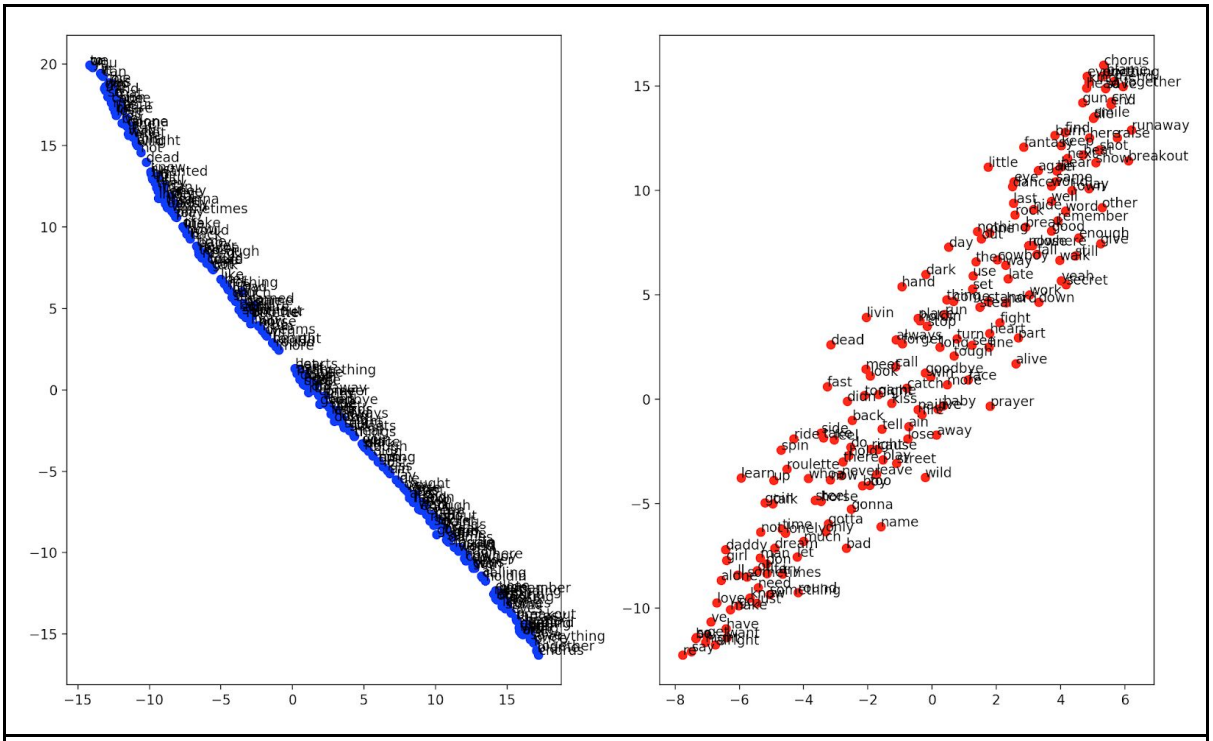

Figure 4: words left, lemmas right

Testing 176 lyrics

Given the conclusions found for several lyrics the same parameters have been applied for the 176 lyrics from the 15 Bon Jovi albums, which are shown in Figure 5 below:

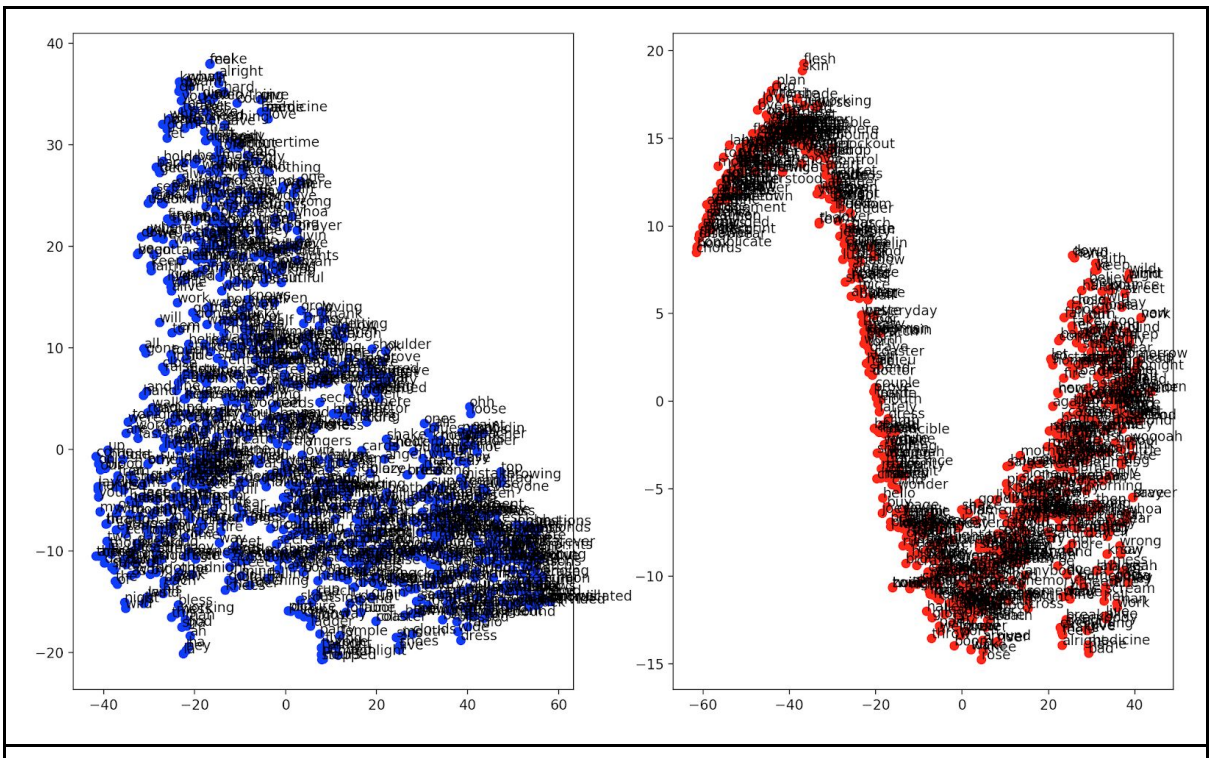

Figure 5: words left, lemmas right

The plots show that the are spots with more density of words which would correspond to the words that are more related and more used in the lyrics, which appear in different contexts.

If only the most common words and lemmas are taken (those that appear more than 20 times in the lyrics for example), the plots show that the names of the lyrics are close, like "*bad medicine*", "*wild in the streets*" and "*diamond ring*" to name a few. Typical words that tend to appear in a lot of songs, are there, for instance: *man*, *woman*, *guitar*, *yeah*, *song*, *tonight*, *goodbye*… This is shown in Figure 6 below.

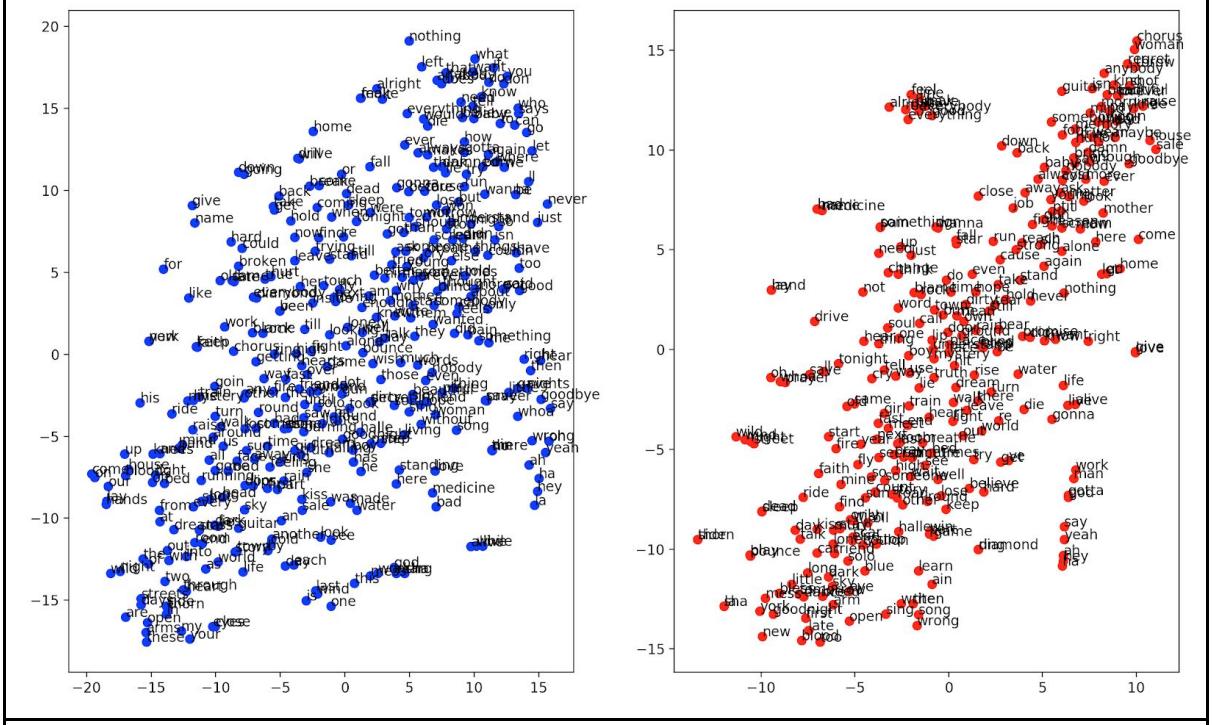

Figure 6: words left, lemmas right

Computing the most similar words/lemmas, funny things can be found, as man related to god, woman with dance, love is something related to give, need or write to, kiss related to dance and first (kiss)... It can be seen that the topics for the songs remain the same as years go by:

- man: ('god', 0.9844035506248474)
- woman: ('dance', 0.9994027018547058)
- love: ('give', 0.994389533996582), ('need', 0.9888852834701538), ('write', 0.9852951765060425)
- kiss: ('dance', 0.9988096952438354), ('first', 0.9986058473587036)

Conclusions

As proven with the plots showed in this work, given a vocabulary constructed from any domain posible, the words/lemmas that are close in the plot are the words that are more related in that domain, those that interact the most in that domain. This is domain dependant, as two words that are related in this Bon Jovi lyrics' domain may not be related in another domain.

Future Work

From here, having the word vectors, we could compute how similar two sentences are or retrieve relevant documents in information retrieval, for instance. Having the semantic relation between the words in the vocabulary, semantic operations can be done, such as finding synonyms or antonyms for one word or sentence.

Git Repository

In https://github.com/ovalls/mai_dl you can find the code for the RNN along with some scripts:

- `bjlyrics.txt`: Bon Jovi lyrics.
- `lab3.py`: code for the creation of the word vectors.