Word Vectors for 244 Countries from Tweets for 300 Spanish Dialects Using Factored Multiskipgram Model

Anonymous ACL submission

1 Introduction

In the field of NLP there exists common problems in syntactic issues such as part-of-speech tagging, chunking, and parsing. While there are also problems in semantic information understanding such as word sense disambiguation, semantic role labeling, and named entity extraction(Tom Young, 2018). As a result word embedding techniques such as skipgram models have began to rise in popularity as they have a means to transform words/phenomes into mathematical vectors for feature extraction.(Oren Barkan, 2016). Moreover, word2vec has seen some use in language translation(Jansen, 2018). In this work, the author does multiple translation between English, German, Spanish and French using word2vec to represent words in a higher-dimensional space. This was done by creating transformation matrices as a way to translate from one model to another. As a result, they were able to obtain candidates for words and phrases. Our work is similar in that we look to translate between different types of "languages" but instead, we are looking to translate between different Spanish dialects.

In different dialects of Spanish, words that seem harmless in one dialect can be quite different in another. In Cuban Spanish `cojer la guagua` can mean `to catch the bus` in English, but in Chilean it means `to fuck the baby`. The word `manzana` in standard Spanish means `apple`, but in Peninsular Spanish, it means `city block`. While `colectivo` means `collective` in most cases, in Argentine spanish it means `bus`. For many of these words the difference have been caused by "loanwords" from different regions. A loanword are any words that is borrowed from another language when it does not currently exist in the language. Detecting and classification of these type of nuances across different regions that Speak spanish is the goal of this work using a novel factorized multiskipgram model. This research is still under going development.

2 The Factorized Multiskipgram Model

We can describe this as the minimizing the following equation:

$$\frac{1}{T_l}\sum_{t=1}^{T_l}\sum_{-c\leq j\leq c, j\neq 0} p_l(w_{t+j,l}|w_{t,l}) \qquad (1)$$

such that

$$p_l(a|b) = \frac{\exp({v'_{a,l}}^T v_{b,l})}{\sum_{w=1}^{W} exp({v'_{w,l}}^T v_{b,l})} \qquad (2)$$

where now we are given L sequence of words that belong to a certain language. Now we index each word with the language as w_{tl} which denotes the t^{th} word of the l^{th} sequence, T_l denotes the total length of the l^{th} sequence, $v'_{w,l}$ is the input vector and $v_{w,l}$ is the output word embedding. Just as in regular skipgram, negative sampling is used to approximate equation 4. We can then calculate a rotation matrix, $A_l : \mathbb{R}^{dxd}$ for each language l such that $A_l v'_{w,l}$ is the final word that is aligned for the word w and language l.

Wen combines equation 3 and 4 into a single step of calculating each models skipgram and alignment such that

$$\frac{1}{l}\sum_{l=1}^{L}\frac{1}{T_l}\sum_{t=1}^{T_l}\sum_{-c\leq j\leq, j\neq 0} \log p_l(w_{t+j,l}|w_{t,l}) \qquad (3)$$

p_l remains the same soft max function in equation 4. Equation 5 arises from the fact that each language's optimization is independent of each other.

Word alignment is done using an average word vector for our word w is defined as

$$\overline{v}_w = \sum_{l=1}^{L} v_{w,l} \tag{4}$$

where an $L2$ penalty

$$\sum_{w=1}^{W} \sum_{l=1}^{L} |\overline{v}_w - v_{w,l}|^2 \tag{5}$$

is used to ensure that each language represents the same words w by ensuring that the vectors are similar. It can be seen from equation 7 that when the vectors $\overline{v}_w$ and $v_{w,l}$ are far apart, the penalty increases.

Our next contribution was to then use a factorized matrix to capture the dependence between word sequences of different languages:

$$v_{w,e} = \nu_{w,e} \cdot \omega_{e,l} \tag{6}$$

where ν is a $A \times W \times E$ tensor and ω is a $E \times D$ matrix and $\cdot$ denotes tensors contraction along the axis e. This factorization allows us to have significantly fewer parameters to learn. This as seen when $E \ll D$. A is our rotation matrix, W is the vocabulary size, E is the number of dialects chosen a priori, and D is a factorization from the $L2$ norm.

3 Training

Our corpus consisted of a Twitter dataset containing 244 different country codes with a total of 48,622,249 Spanish tweets. There is a bias towards countries that have Spanish as their native language and have a larger populatoin with access to the internet as seen in Table 2. This data set is particularly interesting because it is based on phrases and words that are used on Twitter. This is more likely to be inline with how more of a native Spanish speaker might write in a more casual setting. This is different than more formal datasets such as the Billion Spanish words dataset as that scrapes more formal text.

Our model was trained using stochastic gradient descent with a vocabulary size of 50,000 and with an assumption that there are only 300 principle dialects of Spanish which is our parameter E. However, countries like Mexico can have more than a single dialects as there are 32 individual states. In the future we look to expand and find the most optimal number. We used a sampling size of 250

Source Word (CH)	Tar. Dialect	Translation
uno	cl	uno
uno	cu	uno
cojer	cl	to fuck
cojer	cu	to catch

Table 1: Desired/expected results after training

Country Code	Tweets	Words
AD	7640	71681
AR	11378502	11378502
ES	7995187	7995187

Table 2: Sample of dataset for Andorra, Argentina, and Spain

when using negative sampling, and have not yet found a set of hyper parameters that optimizes for best results yet.

4 Desired Results

Before we can capture synonyms in dialects, we must ensure that our model translates semantically the same on common words such as uno, dos, tres as seen in Table 1. But we also expect our models to be able to catch subtleties like cojer.

References

Stefan Jansen. 2018. Word and phrase translation with word2vec.

Noam Koenigstein Oren Barkan. 2016. Item2vec: Neural item embedding for collaborative filtering.

Soujanya Poria Erik Cambria Tom Young, Devamanyu Hazarika. 2018. Recent trends in deep learning based natural language processing.