**Vector-based Machine Translation between English and German**

**Jack Beckitt-Marshall, Class of 2021**

This summer, I explored the use of word2vec, an algorithm developed by Tomas Mikolov et. al at Google. Word2Vec uses a neural network to essentially learn how words and their relationships can be represented mathematically, where words that are more similar to each other have closer distances to one another.[1] My project aimed to explore how word2vec could be applied to translation between the English and German languages, in order to allow the use of linear algebra techniques to translate between the two languages.

The first step of this project was finding a large body of texts to enable accurate training of the word2vec model. There are countless online sources of texts that I could harness for my project, such as Project Gutenberg and Wikipedia. In addition, I also utilized the Google 5-grams dataset, containing sequences of 5 words found in Google's repository of books. The corpuses themselves are extremely important, because they are where the word relationships are deduced from. Hence, it was important to ensure a wide variety of texts were used, as I wanted to avoid overspecialization on certain types of text – if I only used scientific texts, for example, the model would lack the correct relationships that would make it useful for more general texts.

It was also important to preprocess the texts before training my word2vec model, as if we were to train on the raw texts, it could pose issues such as ambiguity: does the word "bear" refer to a large carnivore or the act of supporting something, for example? Therefore, it was important that we analyze texts to ensure that each word has a corresponding part of speech. To perform this, I used the spaCy natural language processing library with the Python programming language. I was able to lemmatize (create the generic form of) words and give each word its corresponding part of speech within the sentence. This was a challenging process at times due to the large amount of texts that required processing (around 250 gigabytes in total) and the fact that the processing itself requires a lot of computational power. As lemmatization is one of the key problems in computational linguistics, the results can often be imperfect too.

There were other differences between the two languages that needed to be addressed, namely German's famous compound words, which can comprise many words that are joined together. These can include short, well-known examples like "schadenfreude" (damage pleasure), but can comprise many individual words, such as the famous "Rhabarberbarbarabar" (Rhubarb Barbara's bar). Splitting compound words was a task I wanted to accomplish as it would aid translation into English, which does not use them to the same extent. I eventually used a variation of Swanepoel and Fick's algorithm for splitting compound words in the Afrikaans language (a Germanic language which is a close relative of German). The eventual algorithm finds the most probable splits using the spaCy library, then it sees whether there are further probable splits using the results.[2]

I later used Bowdoin's HPC (high-performance computing) grid to train the word2vec model. Having a model which has vector representations of words in both English and German is very useful as it can lead to interesting observations – we can examine distances between words as well as other properties that only a vector-based model can deliver us, and we can observe helpful patterns in translation.

The creation of the model was a springboard for further research, such as finding the dependencies between words of both English and German sentences using spaCy. An interesting observation I made was that the parse trees in both English and German were broadly similar. This provided me with an impetus to try out several techniques to enhance machine translation, such as finding how words are ordered in each language based on their dependencies, which can them be used to find rules for translation between the two languages. The main project I have been working on involving the model is enhancing the accuracy of the translation matrix between the two languages, by using the dependencies to find potential English-German pairs of words. To evaluate these pairs, I came up with a process of "word triangulation", looking at their relative distances and gradients when compared to three very common English words and their German equivalents. In the future, I wish to explore other possibilities, such as using sentence examples to learn rules and grammar patterns.

**Faculty Mentor: Eric Chown**

**Funded by the Kaufman Family Fellowship**

---

[1] Tomas Mikolov et al., 'Distributed Representations of Words and Phrases and Their Compositionality' (Mountain View: Google Inc., 2013).

[2] Tilla Fick and Chris Swanepoel, 'Recursive Decompounding in Afrikaans', in *Text, Speech and Dialogue*, ed. Ivan Habernal and Václav Matoušek (Springer Berlin Heidelberg, 2011), 251–58.