1 Background

Introduction and Background

Information retrieval
Information Retrieval(IR)is the activity of gathering relevant information, given another initial piece of information. The most practical example of this is a search engine. Given one or more search words the search engine will attempt to find relevant information. For example, a Google search on "Information Retrieval" (initial piece of information) will give you a list of results (relevant information). To be able to do this, the computer (search engine) must know which texts are related. To achieve this, multiple techniques can be used, such as TF-IDF, Word2Vec, Paragraph Vectors and GloVe. These techiques can be divided into two categories, ones that use a neural network and ones that do not.

Neural Network
Neural Networks are computational structures, based on vector and/or matrix calculations. They can be applied to many different tasks, but need to be trained for each task. This training is the process of iteratively adjusting multiplication matrices or vectors to achieve the optimal result. Which is the result that is as close as possible to the given output for different input and output sets. In some IR techniques, Neural Networks are used to either predict words that may occur around a given word, or predict a word given words that surround it. For example, given the sentence

"The search engine searches for relevant information"

the Neural Network can be trained to either predict the words around "searches" (the, search, engine, for, relevant, information) or to predict the word "searches", given the surrounding (the, search, engine, for, relevant, information). This trained matrix (vector per word) now indicates "word relatedness" which, as mentioned earlier enables association (thus retrieval) with related(relevant) texts.

Embedding
The created vector for a word is referred to as a word embedding. An embedding is a distributed, numerical representation of a text which can capture both the semantic and syntactic information[**?**]. In the case of a word embedding, the embedding represents the word. The advantage of the machine learning models that create these embeddings is that they do not need human interaction **?**], they are so called "unsupervised learning algorithms". Once trained, the embeddings can be used to construct embeddings of larger texts. For example, a word embedding can be used to create a sentence, paragraph, document or corpus embedding. The usage of word embeddings have improved various Natural Language Processing areas such as named entity recognition, part-of-speech tagging, parsing, and semantic role labelling **?**].

Text analysis techniques
To enable a computer to process text, for embedding creation or for tasks, the text has to be processed by an algorithm. In this research we used embeddings created by Word2Vec and TF-IDF feature vectors.

Word2Vec
Word2vec word embeddings are created using neural network, Word2vec learns word embeddings via maximizing the log conditional probability of the word given the context word(s) occurring within a fixed-sized window. Therefore the learnt embeddings contain useful knowledge about word co-occurrence[**?**]. There are multiple input/output possibilities for the neural network, best known are Skip-gram and the Continuous Bag-of-Words model (CBOW). The Skip-gram model takes a target word as input and outputs the predicted output words, while CBOW takes the context words as input and outputs the predicted target word[**? ?**].

Paragraph vectors
Variations on the word2vec model have also been proposed, **?**] introduced the paragraph vector, based on the word2vec model. The paragraph vector model uses additional variables to improve the accuracy of the word-embeddings. An advantage of the paragraph vector model is that it takes the word order into consideration, atleast in a small context [**?**].

GloVe

?] introduced the GloVe (Global Vectors) model. This model captures the global corpus statistics. The model transforms the word co-occurrences of all words in the corpus to chances, it excludes all the zero values and uses that as initial input for the neural network.

TF-IDF

TF-IDF is an abbreviation for Term Frequency times Inverted Document Frequency. This method does not rely on a neural network and therefore does not require training. The TF-IDF score is the product of the term frequency in a text and the inverted document frequency of the same term in a corpus of texts. Both of which can be calculated in a variety of ways. The feature vectors produced by TF-IDF do not capture syntactic or semantic information about words, but capture information about word occurrences.

Validation methods

The results produced by the techniques have to be validated to determine their quality (in usage). The quality of the results can be validated through, among others the F1 score and, for classification tasks, the rank of the correct category. The embeddings can furthermore be validated through their performance on tasks such as word analogies, word similarities, categorization and position visualization. These tasks can be designed to produce a score that gives an indication of the performance on a specific task. **?**] found that a single validation metric cannot produce a representative result for other tasks. Embeddings that perform well on one task do not have to perform well on another task. As a result, the findings about performance of an embedding method are limited to the task on which they are tested, their results cannot be generalized to state that the embedding are overall "performing well". Validation tasks use either labelled or unlabelled data. Labelled data is data that is in some way marked, so that the correct answer can be derived from it, in contrast to unlabelled data.

Word Analogy

Word analogy validation is based on a labelled validation set, containing, commonly, word pairs of four, that can be logically divided into two parts. As Table **??** shows, each last word can be derived from the three words before. The score is the fraction of correctly given fourth words, given the first three words. This validation metric is used in multiple studies[**? ? ? ?**].

Man	Women	King	Queen
Athens	Greece	Oslo	Norway
great	greater	tough	tougher

Table 1: Word analogies examples

Both this validation technique and the Word Similarity technique use vector distance calculations to validate the embeddings, this can therefore also be written as:

$$\mathrm{X}_{\mathrm{Man}} \text{ - } \mathrm{X}_{\mathrm{King}} \approx \mathrm{X}_{\mathrm{Women}} \text{ - } \mathrm{X}_{\mathrm{Queen}}$$

This means that the resulting vector of embedding of "Man" minus the embedding of "King" is approximately the embedding of "Woman" minus the embedding of "Queen". This resulting vector may be close to a vector "monarch" for example.

Word Similarity

A method to test the quality of word embeddings is the word similarity test. For these test, the distance between the word embeddings (vectors) is measured and compared to similarity scores defined by humans. Multiple non domain specific validation sets are publicly available including: the Rare-word dataset introduced in the paper "Better Word Representations with Recursive Neural Networks for Morphology" by **?**], the MEN test collection by **?**] and the WordSimilarity-353 test collection by **?**]. These sets, among others, have been used in multiple studies of word embeddings[**? ?**]. This validation metric also relies on labelled data.

Classification

A classification validation method is a simple task which assigns a label to a text. **?**] used data from StackExchange and tried to determine if a pair was a duplicate. In their setup, the text was a pair of texts, and their categories were duplicate and non-duplicate. **?**] used for their research a dataset of IMDB with 100,000 movie review. They validated their proposed paragraph vector model by determining whether a review was positive or negative.

Position Visualization
?] and **?**] mapped their word embeddings from a high dimensional vector to a two dimensional vector to be able to display them in a scatter plot and applied colors to various categories. These categories can be created with labelled or unlabelled data. The advantage of this is that a human can directly see the word distributions, and see if it is distributed in a way that seems logical. It gives furthermore insight in the overall spectrum of the words. However, this representation does not give a score, since it is not a evaluation of the data, but an alternative representation.

General and Domain specific
Since the word embeddings are created form a given text, these embeddings are bound to the text. All meaning embedded in the word embedding is derived from the original text. Because of this, embeddings can be "Domain specific" meaning that it only knows words (or a specific word) in a certain context. This becomes most clear when faced with words that can have different meanings in different contexts. For this research we categorize the embedding in two categories, general embeddings and domain specific embeddings. The general embeddings are trained on a collection of texts that use common English and contains a wide variety of topics. The domain specific embeddings are trained on a collection of texts that uses uncommon English (i.e. domain specific terms) and/or is limited to a small amount of topics. Given these terms, we regard the embeddings trained on the Wikipedia corpus[**? ? ? ? ?**] as general, since Wikipedia uses common English and spans a wide range of topics. On the other hand we regard the embeddings created by **?**] as Domain specific, these embeddings were created on academic articles, which use domain-specific terms and notations and only consists of academic texts, which contains less general/generic words compared to the Wikipedia corpus.