Thesis plan
Version 1.0

Arjan Meijer
11425555

April 6, 2018

Contents

1 Introduction

This thesis will be about document embedding, embedding, in this context, means: "A method which converts a piece of text, such as title, abstract or an entire document, to a multidimensional numerical vector"[1]. The current works on document embedding focusses on finding new, or optimizing existing embedding methods and applying these embedding vectors for (new) tasks. This work is based on publicly available everyday-language texts (like Wikipedia and forums). While the word-to-vec model, which produces word embeddings, works for most text, researchers encountered problems with the technique when using text that contained sarcasm[7]. This exception shows that the model is not without it flaws. Furthermore, the word-to-vec model can be trained with varying settings/variables, resulting in different levels of accuracy[2]. The goal of this thesis is to apply the word-to-vec model to academic texts, which differ in layout and the usage of language from the everyday-language texts , to find optimal the optimal settings/variables for these texts and to detect possible anomalies. This thesis will be done in cooperation with Elsevier, which has a large collection of scientific papers that can be used for this thesis.

[1] As given by Dr. R. Karimi (Elsevier)

[2] Embedding accuracy can be measured by its usefulness for the task at hand. A good embedding will perform greatly for many various tasks. Generally, a good embedding will map similar texts to close-by vectors as long as they are semantically closely related. Moreover, ideal embedding will create vectors that can be used algebraically such that summation and differences maintain semantic relations.

2 Project Information

Project title:
Embedding for scientific texts.

Student name:
Harm Jan (Arjan) Meijer.

Host organization:
Elsevier (Amsterdam).

Contact information Elsevier:

Name	R. Karimi
Title	Dr.
Function	Lead Data Scientist in Search & Data Science Division
Mail	r.karimi@elsevier.com
Telephone	+31 (0)20 485 3514

Contact information Student:

Name	H.J. Meijer
Title	Ing.
Function	Internship/Student
Mail	meijerarjan@live.nl
Telephone	+31 (0)6 3388 5147

Terminology[3]:

Embedding:
A method which converts a piece of text, such as title, abstract or an entire document, to a multidimensional numerical vector

Accuracy:
Embedding accuracy can be measured by its usefulness for the task at hand. A good embedding will perform greatly for many various tasks. Generally, a good embedding will map similar texts to close-by vectors as long as they are semantically closely related. Moreover, ideal embedding will create vectors that can be used algebraically such that summation and differences maintain semantic relations.

[3]As given by Dr. R. Karimi

2.1 Project Summary

This project looks at the usefulness of embedding for academic texts. More specifically, can embedding methods have a higher accuracy than the more traditional TD-IDF approach. This topic is closely related to machine learning, since the document embeddings are obtained through machine learning. This project produces an overview of the accuracy of embedding methods and the TF-IDF method for representing academic texts. With this information, researchers or people from industry can make a rational decision about using an embedding method for academic texts. This project gives an answer to three research questions:

- Has the word-to-vec embedding method a higher accuracy for academic texts than TF-IDF?

  Which parameter values create embeddings with the highest accuracy for academic texts?

- What is the relationship between the accuracy of an embedding method and the type of text?

To answer the first research question, the optimal parameters for the embedding method have to be found. The comparison between embedding and TF-IDF must only be done with the optimal parameters for the embedding method.

3 Problem Analysis

Researchers have shown in previous studies that text related tasks, like search or recommendation, can be accomplished with embeddings of texts. There are a number of different embedding methods proposed. In this research, I want to see if embedding methods have a higher accuracy than the more traditional approach, TF-IDF. I want to do this for the domain of academic texts, which differ in their document-build up and language from earlier researched online sources, forum like sites and Wikipedia. Elsevier has a large set of academic texts and already applies different embeddings to these texts. Their collection and experience on this topic will enable me to do this research.

4 Research method

4.1 Method & Validation

Method To answer the stated research questions, a fitted approach must be applied. For this project, the research method '*Controlled Experiments*' will be applied. Using this methodology, the goal of this project is to validate these two hypothesis:

- The word-to-vec embedding method can have an higher accuracy for academic texts than the TF-IDF algorithm.
- The accuracy of academic embedding methods can be validated using categorization tasks

Validation This research will be validated via a clear presentation of all compared results. The usage of a publicly available collection and via a clear presentation of the used parameters for each method.

4.2 Experiments:

This research exists out of multiple experiments, the usage of the methods on two collections, one containing non-academic texts, while the other consists of academic texts. The results mentioned in this paragraph is the accuracy score on the categorization tasks. Each document in the test set will be labelled except for 20%, these unlabelled documents have to be matched to the right set of documents. The correct matched percentage will be the result score for the algorithm/embedding method.

First experiment The goal of the first experiment is to compare the results of my algorithm to earlier researches by comparing the results for the Wikipedia set[4]. This will ensure that the algorithm I use is comparable, and thus add legitimacy to the research. I expect that this is a small experiment, since it should only recreate already found results.

Second experiment The second experiment is finding the optimal parameters for the set of academic papers. This will be done by fine-tuning the parameters in search for an optimal result, furthermore, the results of this process must be registered for comparison. This fine-tuning process is a calculation-intensive process, because of this, I expect that this experiment will take most of the time. This experiment also includes data gathering for the comparison of the methods.

Third experiment The third experiment is creating the TF-IDF algorithm in the research environment and collecting data on its search results. This experiment will not take much time, since the TF-IDF is a simple algorithm to implement.

Comparison The final step is the comparison of the collected data from experiments 2 and 3, with this information, I will be able to accept or dismiss the hypothesis, and with that, answer the research questions.

4.3 Sources:

Multiple articles of the explanation of the embedding methods will be used to create the embedding methods. These sources will be included in the Bibliography. Furthermore, the example set that is used in the papers will also be used for this project, the link to this publicly available source will be added as a footnote when the collection is used/discussed. The other collection that will be used belongs to Elsevier and is not publicly available. This set will be described (type of texts, size) and it will be mentioned that the collection is not publicly available since it is property of Elsevier.

4.4 Problems:

Difficulties in this project will likely be caused by finding the optimal parameters for an embedding method. This is a hard and not jet solved problem, it can theoretically be achieved with a brute-force-approach. But the training of the embedding methods is a computational-expensive method. A brute-force over multiple parameters is therefore a very computational-expensive process. A machine learning solution for this process would be a batch-approach. This approach would take a small set of data and use it to predict the (semi) optimal parameters for larger sets. The problem with embedding is that the parameter settings are not scalable, parameters like corpus-word-size can influence the learning result in unexpected ways due to the variation of this value in larger a corpus(i.e. a word corpus size of 100 may be sufficient for a small subset of texts, but may be insufficient for the entire set).

5 Expected project results

This research results in the following findings:

- Comparison of multiple embedding methods for a collection of academic texts

  - If possible, a best-to-use method

- Validation of the hypothesis 'The accuracy of embedding methods is related to the type of text.'

  - If these are related, an analysis of relationship in accuracy of the methods for the academic and non-academic texts.

[4]Which is also used in earlier researches

6 Required expertise for this project

This project requires:

- Basic understanding of Machine Learning
- Knowledge of Databricks/Spark (AWS)
- Knowledge of embedding (methods & evaluation)
- SQL programming
- Python programming

7 Timeline

Focus	**Number of weeks**	**Date**
Setting up workspace	1	April 3 - April 6
Implementing methods	2	April 9 - April 20
Finding optimal embedding parameters	3	April 23 - May 11
Data gathering experiment 1	1	May 14 - May 18
Data gathering experiment 2	1	May 21 - May 25
Writing	2	May 28 - June 8
Correction	1	June 11 - June 15
Submission	1	June 18 - June 22
Total	12	presentation date (NOT YET CONFIRMED) July 10

* for non-academic texts
** for academic texts

8 Risks

Risk	**Prevention / Solution**
Unavailability of previously used datasets	The dataset is already on an local PC
Problems with implementing the methods	Elsevier has a range of experts on this topic, who are willing to help out if needed.
Due to unforseen problems, there is not enough time to complete the entire research	The research consist of multiple hypothesis, atleast one must be achievable
Student has a knowledge gab	I am taking an online course to understand basis of Machine Learning and to get used to Databricks

9 Literature survey

The literature survey provides the explanation of the embedding methods that will be compared in this research. Furthermore, they provide non-academic texts[5] that can be used to validate the implementation of the methods. This set will also be used to compare the non-academic texts to the academic texts.

9.1 Deep Neural Networks for YouTube Recommendations

The paper 'Deep Neural Networks for YouTube Recommendations' by Covington et al.[2] describes the current (16 September 2016) YouTube candidate generation and ranking neural networks. The candidate generation network works with limited data inorder to reduce the needed time for the candidate generation. This process reduces the amount of possible videos for the ranking network, which uses more parameters to increase the accuracy of the ranking. The paper also discusses variables to influence the results of the networks, such as age of the training example and watch-time vs click-through prediction. The paper also discusses the impact that layers of depth have at the results.

9.2 Distributed Representations of Sentences and Documents

The paper 'Distributed Representations of Sentences and Documents' by le et al.[7] proposes the Paragraph Vector as a replacement for the popular bag-of-words or bag-of-n-grams. The Paragraph Vector framework is based on the word vectors framework. The difference between the frameworks is the calculation of the probability, the Paragraph Vector framework uses an matrix D, which consists of every paragraph. This matrix is used to replace a concatenation or average of word vectors. The Paragraph Vector out preforms the state-of-the art techniques on several text classification and sentiment analysis tasks.

9.3 Distributed Representations of Words and Phrases and their Compositionality

The paper 'Distributed Representations of Words and Phrases and their Compositionality' by Mikolov et al.[9] presents several extensions that improve the quality of vector training and its speed. This is achieved by introducing Hierarchical Softmax, Negative Sampling and the subsampling of frequent words to the Skip-gram approach. This does not only speed up the learning process, but also improves the quality for rare words.

9.4 Document Embedding with Paragraph Vectors

The paper 'Document Embedding with Paragraph Vectors' by Dai et al.[3] describes a larger/more thorough evaluation of the Paragraph Vectors model. The writers compare this technique with the Bag of Words model and the Latend Dirichlet Allocation model. The paper states that the Paragraph Vectors model preforms well for grouping, triplet finding and related object/article finding tasks. The paper states that the Paragraph Vectors model is significantly better then the other models in these tasks on the Wikipedia and arXiv articles.

9.5 Stochastic Neighbor Embedding

The paper 'Stochastic Neighbor Embedding' by Hinton et al.[5] describes a probabilistic approach to transform objects form a high-dimensional space to a low-dimensional spaces so that neighbour identities are preserved. To achieve this the objects are transformed to images. A Gaussian is then used to define the (context) probability. This method has the ability to be extended to mixtures in which ambiguous high-dimensional objects can have several widely-separated images in the low-dimensional space.

[5]The papers contain references to publicly available sets that they used for their research.

9.6 An empirical evaluation of doc2vec with practical insights into document embedding generation

The paper 'An empirical evaluation of doc2vec with practical insights into document embedding generation' by Lau et al.[6] empirically evaluates the quality of the documents embedding created with the method 'doc2vec' compared to 'word2vec' and the 'n-gram' model. They conclude that doc2vec preforms well and that the 'dbow' approach works better than the 'dmpv' approach for the doc2vec method. They also provide recommendations for the optimal parameters for the doc2vec method.

9.7 Efficient estimation of word representations in vector space

The paper 'Efficient estimation of word representations in vector space' by Mikolov et al.[8] shows that, with newly applied techniques, the size of the training sets can increase while the time needed for the training operation does not increase. This new approach greatly reduces the training time per word and allows for bigger sets of texts. This results in a higher accuracy of the created embeddings. The authors state that the newly applied techniques can also be used on other, already existing embedding methods.

9.8 GloVe: Global Vectors for Word Representation

The paper 'GloVe: Global Vectors for Word Representation' by Pennington et al.[10] proposes a new model, named GloVe, which is a log-bilinear regression model for unsupervised learning of word representations. This model out preforms other models on three specific tasks, these are: word analogy, word similarity and named entity recognition.

9.9 MapReduce: Simplified data processing on large clusters

The paper 'MapReduce: Simplified data processing on large clusters' by Dean et al.[4] explains the MapReduce programming model. The model is inspired by functional programming languages. It requires programmers to specify a map function which processes key-value pairs to generate intermediate key-value pairs. It requires furthermore a reduce function, that merges all intermediate values associated with the same intermediate key. This method allows for an execution on a large distributed computer network. This allows the processing of large input (and output) sets in reasonable time.

9.10 Spark SQL: Relational data processing in spark

The paper 'Spark SQL: Relational data processing in spark' by Armbrust et al.[1] explains Spark SQL. Spark SQL lest programmers leverage the benefits of relational processing and lets SQL users call complex analytix libraries in Spark. This allows for much tighter intergration between relational and procedural processing. The paper further states that Spark SQL makes it significanly simpler and more efficient to write data piplines that mix relational and procedural processing, while offering substantial speedups over previous SQL-on-Spark engines[6].

[6]Based on user feedback

References

[1] Michael Armbrust, Reynold S Xin, Cheng Lian, Yin Huai, Davies Liu, Joseph K Bradley, Xiangrui Meng, Tomer Kaftan, Michael J Franklin, Ali Ghodsi, et al. Spark sql: Relational data processing in spark. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1383–1394. ACM, 2015.

[2] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 191–198. ACM, 2016.

[3] Andrew M Dai, Christopher Olah, and Quoc V Le. Document embedding with paragraph vectors. *arXiv preprint arXiv:1507.07998*, 2015.

[4] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[5] Geoffrey E Hinton and Sam T Roweis. Stochastic neighbor embedding. In *Advances in neural information processing systems*, pages 857–864, 2003.

[6] Jey Han Lau and Timothy Baldwin. An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*, 2016.

[7] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196, 2014.

[8] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[9] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[10] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.