The Language of Engineering

Training a Domain-Specific Word Embedding Model for Engineering

Daniel Braun
Technical University of Munich,
Department of Informatics
daniel.braun@tum.de

Oleksandra Klymenko
Technical University of Munich,
Department of Informatics
alexandra.klymenko@tum.de

Tim Schopf
Technical University of Munich,
Department of Informatics
tim.schopf@tum.de

Yusuf Kaan Akan
ROKIN GmbH
kaan.akan@rokin.tech

Florian Matthes
Technical University of Munich,
Department of Informatics
matthes@tum.de

ABSTRACT

Since the introduction of Word2Vec in 2013, so-called word embeddings, dense vector representation of words that are supposed to capture their semantic meaning, have become a universally applied technique in a wide range of Natural Language Processing (NLP) tasks and domains. The vector representations they provide are learned on huge corpora of unlabeled text data. Due to the large amount of data and computing power that is necessary to train such embedding models, very often, pre-trained models are applied which have been trained on domain unspecific data like newspaper articles or Wikipedia entries. In this paper, we present a domain-specific embedding model that is trained exclusively on texts from the domain of engineering. We will show that such a domain-specific embeddings model performs better in different NLP tasks and can therefore help to improve NLP-based AI in the domain of Engineering.

CCS CONCEPTS

• **Information systems, Information retrieval, Retrieval tasks and goals, Document filtering**; • **Information systems, Information retrieval, Retrieval tasks and goals, Clustering and classification**; • **Computing methodologies, Artificial intelligence, Natural language processing, Language resources**;

KEYWORDS

Word Embeddings, Engineering

ACM Reference Format:
Daniel Braun, Oleksandra Klymenko, Tim Schopf, Yusuf Kaan Akan, and Florian Matthes. 2021. The Language of Engineering: Training a Domain-Specific Word Embedding Model for Engineering. In *2021 3rd International Conference on Management Science and Industrial Engineering (MSIE 2021), April 02–04, 2021, Osaka, Japan.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3460824.3460826

1 INTRODUCTION

Digital platforms such as booking.com or Amazon have massively changed the way we live. They make daily routines efficient and pleasant by helping us to quickly access the right information, no matter where we are. However, there is not one digital platform that serves all our needs, instead, we use different specialized platforms for different aspects of our lives: to communicate, order food, book holidays, find a flat or even a life partner.

Most of these information platforms are aimed for a B2C context. When it comes to B2B and engineering information, no specialized platform and no access to specific information for this domain exists. For information retrieval, to solve their technical challenges, engineers use the same tools today, as 30 years ago. These information retrieval tools consist of trade fairs, trade publications or online research using search engines. One challenge that these tools face is the exponential growth of information in the engineering sector. This growth is a result of a growing number of engineering disciplines as well as a strong growth in digitally communicating the progress that has been made in these various disciplines.

For these reasons, engineers spend on average 16% [1] of their working time searching for information on potential solutions for their technical challenges or how to improve both their product and its production process. During this time of search, engineers are not actively creating any value for their companies. This leads to a financial loss in the German engineering sector of € 18 billion [1].

To fill this gap of information retrieval for engineers, new tools that can cope with handling the continuous increase of information and that understand the information need of an individual engineer are necessary. Modern AI systems that use advanced natural language processing (NLP) algorithms offer the possibility to fill this gap. These algorithms can act as a digital assistance system for engineers, understanding the individual information need and with access to large amounts of engineering information can fit the most relevant content for this need. Unfortunately, in the engineering domain, no specific language models exist that are precisely adapted to meet the requirements of the used technical jargon and terms. This leads to an unsatisfying information retrieval process and still to a lot of manual rework.

In this paper, we present an approach to deal with the specific technical jargon and terms, by training specific engineering NLP models based on a large number of technical texts. With this, we

MSIE 2021, April 02–04, 2021, Osaka, Japan

ACM ISBN 978-1-4503-8888-7/21/04…$15.00
https://doi.org/10.1145/3460824.3460826

are able to show a better understanding of engineering information and an improvement of basic NLP tasks within the domain.

2 RELATED WORK

Word embeddings are dense word vector representations that serve as the main input for a variety of NLP tasks and are therefore of high interest to many researchers.

Most of the current work focuses on domain-agnostic methods, where techniques such as Word2Vec [2] or GloVe [3] are used to train word embeddings on generic text corpora. However, such approach results in general word representations that do not perform as well on domain-specific tasks that have to deal with, for example, legal, technical or medical texts. In order to perform optimally, these tasks require customized word representations that would account for the domain-specific vocabulary and semantic associations.

While to the best of our knowledge, there has been no prior work on constructing word embeddings specific to the field of engineering, some researchers have approached this task for other domains.

Efstathiou et al. [4] trained word embeddings for the software engineering domain using preprocessed text from Stack Overflow[1] posts, demonstrating the ability of their model to capture software-specific semantics, as well as to disambiguate polysemous words such as “cookie” or “smell” to their software engineering meaning.

Chalkidis and Kampas [5] trained legal word embeddings on a large corpus of legislation from various countries and presented a corresponding model they called Law2Vec that was made publicly available for further use. In the work of Risch and Krestel [6], authors presented word embeddings trained specifically for the patent domain that have proved to outperform generic embeddings trained on English Wikipedia articles for the task of patent classification.

Wang et al. [7] compared the performance of different word embeddings in the biomedical NLP, demonstrating that word embeddings trained on clinical notes and biomedical publications are able to find more similar medical terms than the ones trained from Wikipedia and news articles, and that semantic similarity captured by embeddings trained on clinical notes is the closest to the view of human experts.

In the work of Nooralahzadeh et al. [8], authors evaluated embedding models trained on technical reports and scientific articles from the oil and gas domain, showing the effectiveness of the domain-specific approach even with limited training data.

3 DATA CORPUS

The dataset used in this paper consists of text documents that have been collected over several months from more than 100 engineering trade publications in English in the domain of mechanical and electrical engineering. Trade publications with the highest distribution and awareness in the engineering community were selected for the dataset such as www.engineering.com, www.eetimes.com and more. The publications focus mainly on engineering topics such as robotics, automation, 3D printing or augmented reality, but also on more economical aspects such as investments, mergers and acquisitions or Personnel changes in companies.

The text documents were collected directly from the websites. Extraction of raw texts was done using RSS feeds if available or by using specialized crawling algorithms. The crawling algorithms were implemented through Python, using the beautiful soup as well as scrapy framework. In total the dataset consists of roughly 600,000 technical articles from the engineering sector. A crawled text document consists of title, body text, URL and publication date. On average the length of an article body text consists of roughly 1.300 characters. The article’s dates range from 1969 up to August 2020. With the majority of articles being published later than 2010.

Due to the unstructured extraction of websites the articles often contained artefacts from various aspects of the website html structure - such as headings, subsection names, picture, author signatures and more. In order to work with the dataset, it was subjected to a pre-cleaning process in the form of regular expression removal. The pre-cleaning process removed several artefacts and improved the overall data quality to reduce biases on subsequent natural language processing tasks.

4 TRAINING

In order to train the domain-specific embeddings models from the data corpus, we used the Python library gensim[2]. We trained a bigram model with a dimensionality of 300 and a windows size of 5. The training was performed on an Intel i5 Dual-Core processor with 2.9GHz and 16 GB of DDR3 RAM. Building the model took around seven to ten hours on the machine. The resulting model contains vectors for over 1.1 million words and has a size, stored in binary format, of approximately 500 MB.

5 EVALUATION

One of the main promises of word embeddings is that their vectors encode semantic meaning and that the distance between these vectors is a measurement for semantic similarity. The most straightforward approach to compare different word embeddings with each other is therefore to compare how well they encode semantic meaning in a given domain. We will do that by comparing synonyms (or more specifically the most similar words) for a number of words from the domain of industrial engineering and a number of words from other domains. We will then ask human experts to judge how semantically similar the lists of similar words, produced by the different embedding models, really are. Our hypothesis is that large pre-trained word embeddings perform better on words not related to industrial engineering, while the model we trained will perform better on words from the domain of industrial engineering.

Additionally, we will also compare the performance on a real-world classification task: We will classify whether a given article from a trade journal from the domain of industrial engineering describes a new technology or not. Since this is a domain-specific task, our hypothesis is that the word embeddings we trained will perform better on this task than a general-purpose embedding model.

As comparison point, we used the original Word2Vec model [2], which was trained on a Google News dataset which contained about 100 billion words. The resulting model has the same dimensionality of 300 and contains vectors for about 3 million words and phrases.

[1] https://stackoverflow.com/

[2] https://radimrehurek.com/gensim/

Table 1: Statistical overview of the dataset

Parameter	Value
Toal number of extracted articles	602,903
Average article length in sentences	13.21 (max 296, min 6)
Number of extracted publications	102
Average articles per publication	5,910 (max 140k, min 112)
Oldest/newest article date	1969-12-31 / 2020-08-14

Table 2: Semantically most similar words for “ai” in the Word2Vec model and our Engineering model

#	Word2Vec	Similarity	Engineering	Similarity
1	che	0.63	artifical_intelligence	0.93
2	te	0.62	machine_learning	0.89
3	essere	0.59	deep_learning	0.85
4	é	0.59	computer_vision	0.77
5	mai	0.59	conversational_ai	0.71
6	voi	0.59	analytics	0.71
7	tutto	0.58	predictive_analtics	0.70
8	ti	0.58	ai_ml	0.70
9	tutti i	0.58	cognitive_computing	0.70
10	questo	0.58	image_recognition	0.67

5.1 Synonyms

In the first step, we defined a list of words and bigrams, that are relevant and specific to the domain of Manufacturing Engineering, but also Engineering more broadly:

- AI / Artificial Intelligence
- Cloud Computing
- Robot
- Robotics
- Webinar
- 5G
- 3D
- Sensor
- Car
- Vehicle
- IoT
- Cloud
- Manufacture
- Digital

We then searched the ten semantically most similar word or phrases for both models. Table 2 shows the results for the abbreviation “ai”, ordered from most similar (1) to less similar (10). What we can see is that the standard Word2Vec model interpreted “ai” not as an abbreviation for “Artificial Intelligence”, but as an Italian word and hence returns a list of synonyms for this Italian word. Our Engineering model, on the other hand, identifies the abbreviation correctly and returns the full form, artificial intelligence, as most similar word, as well as a list of words which describe subfields of AI, like machine learning or image recognition.

Table 3 shows the most similar words or phrases in both models for the word “robot”. Especially in the first places, Word2Vec performs better here, because it returns semantically more similar words. Our Engineering model, on the other hand, shows that it was trained on the subdomain of manufacturing engineering by returning words that are very important subsets of robots in this specific domain, like “robotic arm” or “gripper”. Towards the end of the list, the Word2Vec model tends towards specific brands and models of robots, which have been a popular subject in news reporting in the past.

In the next example in Table 4, we show the most similar words or phrases returned by both models for the input “5g”. While the results from our Engineering model clearly show that the model is interpreting “5g” as a mobile communication standard, the results from the general Word2Vec model show that the input is interpreted as a weight.

We analyzed all the domain-specific words in the list above in this way and found that our Engineering model returned better results in seven of the fifteen instances. The Word2Vec model only returned better results for the input “robot”, in the other seven instances, none of the models was clearly better. More generally, we also found that the Word2Vec model tended also for other inputs, like “car”, to return specific brands and models, as we saw for robots. This is probably caused by the fact that general interest media tend to write about technical topics based on specific instances of these, rather than on a general, abstract level.

We also tested both models on various words that are not from the domain of Engineering, like garden, children, weather, trousers, and tree. While the results of the Engineering model were generally speaking surprisingly solid (for weather, for example, the five most similar phrases were severe weather, weather conditions, winter weather, storm surge, and weather patterns), it was outperformed by the Word2Vec model on all of these words.

Table 3: Semantically most similar words for "robot" in the Word2Vec model and our Engineering model

#	Word2Vec	Similarity	Engineering	Similarity
1	robots	0.83	robotic_arm	0.82
2	robotic	0.81	bot	0.76
3	humanoid	0.67	humanoid_robot	0.76
4	robotics	0.65	gripper	0.76
5	humanoid_robots	0.64	robots	0.75
6	honda_asimo	0.63	manipulator	0.70
7	autonomous_robots	0.62	collaborative_robot	0.70
8	geckosystem_suite	0.62	microrobot	0.70
9	i_sobot	0.62	humanoid	0.70
10	manufacturer_kokoro	0.62	robotic	0.70

Table 4: Semantically most similar words for "5g" in the Word2Vec model and our Engineering model

#	Word2Vec	Similarity	Engineering	Similarity
1	7g	0.70	3g	0.87
2	2g	0.69	4g	0.87
3	6g	0.68	lte	0.85
4	8g	0.68	5g_networks	0.83
5	1g	0.66	wimax	0.83
6	9g	0.65	4g_lte	0.82
7	4g	0.64	mobile_broadband	0.78
8	##.#g	0.62	mobile_wimax	0.77
9	#.#g	0.62	hspa	0.76
10	###g	0.62	hsdpa	0.76

5.2 Classification

Assessing the similarity of words is rarely used as an end in itself, but usually to facilitate some other NLP tasks. Therefore, we also compared our embedding model in a real-world task, in which we tried to classify whether a given article contains the description of a newly released technology.

For this task, we first annotated a set of 2,126 articles with a label for whether or not they contain a description of a newly released technology. During this annotation, it became quickly evident that we are facing a difficult decision problem that even for humans is not always easy to solve. 488 articles were annotated as describing a new technology and 1,638 articles as not describing a new technology. The dataset was split into a training (80%) and a test dataset (20%). The training dataset was used to train a Multi-Layer Perceptron, for which the input was once encoded using the Word2Vec standard model and once using our custom Engineering model.

For both configurations, we first performed a five-fold cross-validation on the training data to determine the best performing hyper-parameters. For both inputs, we found a network with two hidden layers and 200 neurons in the first hidden layer and 50 neurons in the second hidden layer to perform best. We also found a constant learning rate with tanh as activation function and adam as solver to work best in both cases. Only the batch size differed: For the Word2Vec model, we saw the best performance when using a batch size of 200, for our Engineering model, the performance was better with a slightly smaller batch size of 150.

When we evaluated both approaches on the hold-out test set, we found that the network which used our domain-specific model performed better in the classification task, with a precision of 0.64, compared to a precision of 0.6 for the generic Word2Vec model.

6 CONCLUSION

In this paper, we presented a domain-specific word embeddings model that was trained on a large corpus of more than 600,000 articles from the domain of engineering and manufacturing engineering. As far as we are aware, it is the largest model of its kind in this domain. We compared our model against the standard Word2Vec model, which was trained on the much larger Google News Dataset. Despite this difference in size, we have shown that our model performed better in two domain-specific tasks while being outperformed by the standard model when it comes to other domains.

More generally speaking, we have shown that domain-specific embedding models can improve the performance in different NLP tasks in a specific domain and therefore, instead of using generic models, it might be worth putting in the time and effort to train more domain-specific models.

ACKNOWLEDGMENTS

The authors would like to thank Thomas Kinkeldei of ROKIN for his contributions to this paper.

This work has been supported by funds from the Bavarian Ministry of Economic Affairs, Regional Development and Energy as part of the program “Bayerischen Verbundförderprogramms (BayVFP) – Förderlinie Digitalisierung – Förderbereich Informations- und Kommunikationstechnik”.

REFERENCES

[1] Kollaborative Produktentwicklung und digitale Werkzeuge Defizite heute – Potenziale morgen. Studie der CONTACT Software GmbH, des Fraunhofer IPK und des VDI, 2013

[2] Mikolov, Tomas, *et al.* "Distributed representations of words and phrases and their compositionality." Advances in neural information processing systems. 2013.

[3] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

[4] Efstathiou, Vasiliki, Christos Chatzilenas, and Diomidis Spinellis. "Word embeddings for the software engineering domain." Proceedings of the 15th International Conference on Mining Software Repositories. 2018.

[5] Chalkidis, Ilias, and Dimitrios Kampas. "Deep learning in law: early adaptation and legal word embeddings trained on large corpora." Artificial Intelligence and Law 27.2 (2019): 171-198.

[6] Risch, Julian, and Ralf Krestel. "Domain-specific word embeddings for patent classification." Data Technologies and Applications (2019).

[7] Wang, Yanshan, *et al.* "A comparison of word embeddings for the biomedical natural language processing." Journal of biomedical informatics 87 (2018): 12-20.

[8] Nooralahzadeh, Farhad, Lilja Øvrelid, and Jan Tore Lønning. "Evaluation of domain-specific word embeddings using knowledge resources." Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018). 2018.