Website visitor segmentation and targeting

Dimitar Dimitrov†  
 Department Name  
 University of Amsterdam  
 Amsterdam, Netherlands  
 [email@email.com](mailto:email@email.com)

Chang Li  
 University Supervisor  
 University of Amsterdam  
 Amsterdam, Netherlands  
 [email@email.com](mailto:email@email.com)

Michael Metternich  
 Company Supervisor  
BloomReach  
 Amsterdam, Netherlands  
 email@email.com

Maarten Marx  
 University Examiner  
 University of Amsterdam  
 Amsterdam, Netherlands  
 [email@email.com](mailto:email@email.com)

ABSTRACT

*Dummy Text*

KEYWORDS

Web Mining, Clustering, Classification, Text Analysis, Convolutional Neural Networks.

ACM Reference format:

FirstName Surname, FirstName Surname and FirstName Surname. 2018. Insert Your Title Here: Insert Subtitle Here. In *Proceedings of ACM Woodstock conference (WOODSTOCK’18). ACM, New York, NY, USA, 2 pages.* https://doi.org/10.1145/1234567890

1 Introduction

The impact of digitalization is causing the rise of different discussions and debates on topics like ethics, policy making or social impact among others [21] and although outside of the scope of this paper it is worth mentioning this due to the fact that it shows how interconnected our lives are with technology.

Although the digital era has changed a lot of domains, in this paper we will focus on something specific: e-commerce. E-commerce represents the act of buying or selling goods through the World Wide Web. [22] As suggested by R. Cooley et al a lot of companies rely on the internet to conduct and expand their business. To motivate this statement according to [4], in 2019 e-commerce is expected to be responsible for over three trillion of dollars in sales. This can lead us to the conclusion that shopping is no longer related to going to the actual shop, rather going to the web version of it. Bearing all this in mind we can understand how important it is to focus on making the online experience of the user better by allowing them to find what they need before they even know it – representing the best-case scenario.

By living in a tech-advanced world we generate bits of data with each action we take online. Having a closer look at the one of the Marketing Theories – Consumer Decision Making Process [7]. We can see that it consists of five different steps and the actual purchase is only the fourth of it. The first three steps have to do with the realization of the individual that he or she needs a certain item, followed by the search of similar items and the evaluation of the different items. While performing those three steps, online, the user is generating their so-called digital identity by leaving footprints, such as a search in a website or visiting a certain category page. The digital identity can then be completed with the actual realization of a purchase.

By collecting, usually in the form of log files, the digital identity of their visitors’ companies can segment them into groups. The groups can be based on, for example a certain product group they have searched for. Based on those segments the business can target them in a more personal way by for example improving and adapting the content they are seeing online, known as content management, thus improving their digital experience.

In the book Misbehaving: The making of Behavioral economics, the author Richard H. Thaler, amongst other things discusses the fact that people tend to not follow economic models and logic. In the following article the author continues this thought by arguing that our buying decisions are more affected by emotions, trust and others than by logic. [23] Having this in mind we can assume that adapting the content based on the user preferences might make the individual feel more relaxed and assist in building trust.

The purpose of the text up until now was to build-in the understanding of how important it is to be able to squeeze value of every bit of data out there. This thesis will focus on the extraction of valuable information from the online journey of the visitors. The project took place in a digital company called BloomReach. The company can be labelled as the middleman, who is improving the experience of the end online-user and his relationship with the business. To be more specific the research was built around one of BloomReach products – Experience Manager. The product has to do with giving more power to the company, by allowing them to analyze and optimize content based on audience. The segmentation is mainly done manually by administrators, which we aim at improving. As mentioned earlier one of the ways to store the digital footprint of the user are the log files, which are also the main foundation of this research. However, we just have visited URL’s without any label or class specification, or simply put – raw data. We could, however, extract insightful information through ‘text analysis’. This information can be used as an input for different algorithms or Neural Networks, but due to the lack fo labels or class specifics, the approach will be unsupervised – meaning that we will have to learn from the data, without having prior knowledge.

The rest of the paper will be structured as follows. As a subsection in the introduction we have the research question, which will guide the reset of the paper. Following the introduction, we will cover the past and current research on the topic. In the third section we will dive into our methodology, where we will explain steps taken into transforming the log files into valuable information. The research paper will be finalized with evaluation and conclusion, where the end results will be presented as well as suggestions for future work.

1.1 Research Questions

Bearing the introduction in mind, although broad it gives us an idea of the importance of user segmentation in the e-commerce, which is also the focus of this paper. In this paper we will try to investigate the application of Convolutional Neural Networks (CNN) in unsupervised clustering of textual data extracted from log files. Motivated by this, the main research question of this paper is:

**RQ1: To what extent can the use of Neural Networks outperform k-modes clustering in unsupervised user segmentation, based on keywords extracted from the online journey?**

As the topic can quickly expand and overpass the time limitation of this project, and to improve evaluation, the work was split in two parts, each of which will have it is own set of sub-questions question. The first part will be mainly concerned with CNN, whereas the second part will cover the research from the point of view of the company/sponsor of this project, by making use of their data.

**Part 1: Convolutional Neural Network**

* #What type of Neural Network can be implemented #with the available data in unsupervised manner?
* To what extent is the suggested/implemented approach performing compared to suggested approaches from the literature?

**Part 2: Available Data inhouse**

* How can we evaluate the performance and results with the data inhouse?
* Can we use an open data set to move from unsupervised to supervised approach?

With the advancements in technology companies can use different techniques such as clustering to classify customers. However, in the recent years we can see a significant increase in the published papers, which are discussing the use of Deep Learning and more specifically Neural Networks.

Another motivation for this direction is the fact that this paper will continue a research, which was previously done within the company. In that research they investigated the use of k-means clustering algorithm for user segmentation.

2 Related Work

This section will be split and structured in multiple sub-sections. Each section will provide different part of information related to the idea, motivation and relevance of the project.

2.1  Web Mining

The motivation of this project lies in some of the work of R. Cooley et al. To be more precise, point of interest are two of his papers:

- Data Preparation for Mining World Wide Web. Browsing Patterns

- Grouping Web Page References into Transactions for Mining World Wide Web Browsing Patters

In the first one the take away is that the authors are describing and discussing different Data Mining techniques, amongst which we have association rules and clustering analysis. The later one being the focus of this paper as it emphasizes on the benefits of grouping similar users together, which is initially what clustering stands for. Based on those clusters the company can either develop a marketing strategy or execute one by targeting customers both online and offline

The latter paper is more about understanding the taxonomy of the web. In the paper R. Cooley et al. share their theory that a certain webpage can have one of two purposes for the end user – either navigation page or a content one. However, the main take away from this paper is the content regarding the user transaction and how to extract what is actually relevant. As mentioned in the Introduction, each search or click we make online is leaving a trace in the log files. With this in mind, we should be aware of how to get only the valuable information. They discuss three different modules; however, we will focus only on the so-called Maximal Forward Reference. Although it will be touched again in the Methodology, it is worth emphasizing that it this module is taking into account visitor ID or something similar which distinguishes one user from another, timestamp of the visit and the page. The idea is that a group of consecutive URL visits will finish with a content page and all the pages leading to the end are navigational, thus have less importance.

2.2  Relevant work inhouse

The exact formulation of the research question has to do with a recent research done within the sponsoring company. As the main research question suggests, we aim at comparing the performance of our approach to another one, previously tested with the same data. In the previous approach the aim was to cluster the users, again, based on keywords extracted from the visited URL’s. The extracted keywords were then fed into k-modes clustering algorithm. K-modes is a variation of k-means, which is able to deal with categorical values, as they did not transform them.

In general, unsupervised algorithm, such as k-means, take input vectors based on the dataset and make assumptions about the data. [24] As stated by Aysegul Dundar et al, in Convolutional Clustering for Unsupervised Learning, the classic k-means algorithm finds cluster centroids (centers) that minimize the distance between points, input vectors, in the Euclidean space. Put simply by Andrey Bulezyuk: ‘the objective of K-means is simple: group similar data points together and discover underlying patterns. To achieve this objective, K-means looks for a fixed number (*k*) of clusters in a dataset.’ Cluster here refers to a grouping of data points based on underlying similarities and each cluster has a centroid.

The relevant work within the company will be further touched in the Methodology section.

2.3  Neural Networks

Section 3.1 Data Description will further discuss the structure of the provided data, however for the purpose of this section and to explain our choice of literature it is worth mentioning that the data at hand represents short texts, without any categories or classes attached to them, as well as arguably any context.

Continuing from section 2.2. K-modes takes vectors from the data, what if we can incorporate such simple algorithm together with feature vectors generated from Neural Networks as input data. Although this question is hardly new there are certain points, which need to be taken into account. For example, Dundar et al, among others, suggest an approach which incorporates both k-means and Neural Networks, however they are using images and they have labels, which is the first point to be taken into account. In general, Neural Networks require some sort of labels, which can be used during the training of the network. Simply put, this is how the network will learn which are important features per class/label. Another important point to take into account is the type of Neural Network you will use, which is affected by multiple things such as the data at hand and expected outcome. There are multiple sources and guidelines, such as the work of J.E.Angus – Criteria for choosing the best Neural Network. [25]

Points to take away are firstly the fact that we need a workaround for the lack of labels in regard to the training of the Neural Network and the second point is the type of data, which was mentioned at the start of the subsection. Starting from latter one, as Wen Hua et al. explain in their work, short texts are limiting in the sense that they lack the syntax and grammar used in proper text. Secondly, short texts are lacking statistical information needed for proper use of statistical approaches like topic-modelling and as the authors state, short texts are ambiguous, thus hard to interpret. In our work we are dealing separate words extracted from URL’s. Regarding the unsupervised training of Neural Networks Jiaming Xu et al, propose, to the best of our knowledge, an approach worth investigating. In their paper – Short Text Clustering via Convolutional Neural Networks, they suggest an architecture capable of learning the most important features without the use of any labels and clustering the output vectors via k-means. Bearing this in mind and the fact that our main idea is to investigate whether or not Neural Networks can outperform k-means, on its own, we decided to follow their approach and improve on it if possible.

3 Methodology

*A few sentences about the project being conducted within a company (also mention it is related to their product), thus it is data specific. Maybe describe initial idea to test different approaches but due to the data at hand the direction moved slightly.*

As already mentioned up until now, this research is rather case specific as in its basis it is constructed on data from one of the main products of the sponsoring company – Experience Manager. Although the initial idea was to go over different Neural Networks and see which one is performing the best, the direction of this research moved slightly to focus mainly on Convolutional Neural Networks (CNN). The switch was motivated by the fact that although CNN’s were initially related and known for their performance in image related task, however as Jenq-Haur Wang et al [26] suggest, they are also good at learning local features from words and phrases – which is also the type of data we are dealing with. The above statement from is also evaluated and confirmed in the work of Yoon Kim, who compares the performance of different simple CNN’s on seven tasks, two of which are sentiment analysis and question classification. The proposed method from Yoon Kim improved on 4 out of those 7 tasks. Besides this, the approach used for the training of the Neural Network is based on the work of Jiaming Xu et al. and as we also aim at comparing our results with theirs, we settled for the same Neural Network as them – CNN, however different design.

The rest of this section is split into sub-sections each aiming at providing part of the whole process and will further built-on the content from above.

3.1 Data Description

*- A dummy table from raw source – change user id*

*- Crawling – mention why you do crawling/scrapping (more data), how – scrapy*

As mentioned earlier the available data inhouse was generated from one of the company’s product – Content Management Software. Whenever a client requests this product, the tool is deployed based on their requirements and is also set up in this way. Bearing this in mind, the tool, one of the things stored by the tool are the log files of the interactions of the visitors with the website. Each deployment of the tool might store different amount of data, depending on the initial set up. Through this project we used log files generated by the standard set up of the system. This was done in order to test whether or not it is possible to segment users based on the least amount of data stored per individual.

Initially two datasets were provided, one from the sponsoring company’s website and one from a client website. Both datasets had the same features per user, however the client dataset was unfortunately multilingual, which differed from the initial idea to focus on content in English, similar to the previous research.

This being said, the preferred dataset was generated and based on the usage of the company’s website – bloomreach.com. The data set contains the following information:

1. Unique identifier, which is distinguishing site visits/sessions
2. Unique identifier, which is distinguishing users and is stored in a cookie
3. Location information, which consist of country, city, latitude and longitude
4. Information about the day of the week the visit was made as well as separate Timestamp per activity per user. This allows to follow the path
5. Browser used
6. Referrer page – the page which got the user to ‘current’ page.
   * Example of referrer page can be google.com

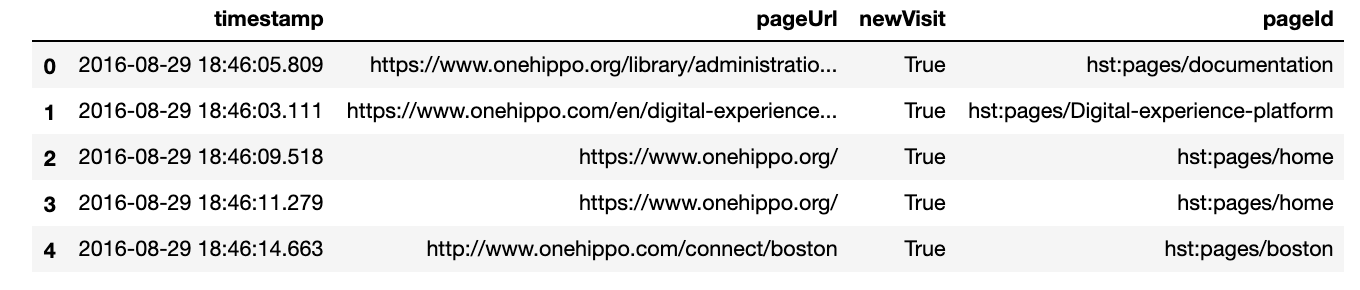
According to the start time stamp, the first record is from 29th of August 2016 and the last record is from 15th of April 2019. In total the log file contains 12 610 512 rows. Based on the visitor identifier there are 6 442 700 unique entries. Filtered sample, without the visitor identifier can be seen in Figure 1. 

Figure 1 Data Sample

3.1.1  Data pre-processing

*All the cleaning steps – FOR EVALUATION maybe compare both cleaning steps with a small random sample. MF function description keep it general no code*

Often log files require certain cleaning in order to be useful as they might contain irrelevant information. Two main points have to be taken from the pre-processing. The first part of the cleaning is to assess and filter out the irrelevant bit of data, such as home pages or other data not relevant to the task at hand. Following this the data is sorted based on visitor and timestamp in order to get the right sequence. As described in the Related work section, single URL visit might lack enough valuable information, but grouping those visits all together will allow one to see the what the visitor was looking for and from there build a valuable visitor feature.

The second part is to clear the URL’s in the transactions and extract the key words from them, per user. Figure 2 represents an example of a URL.

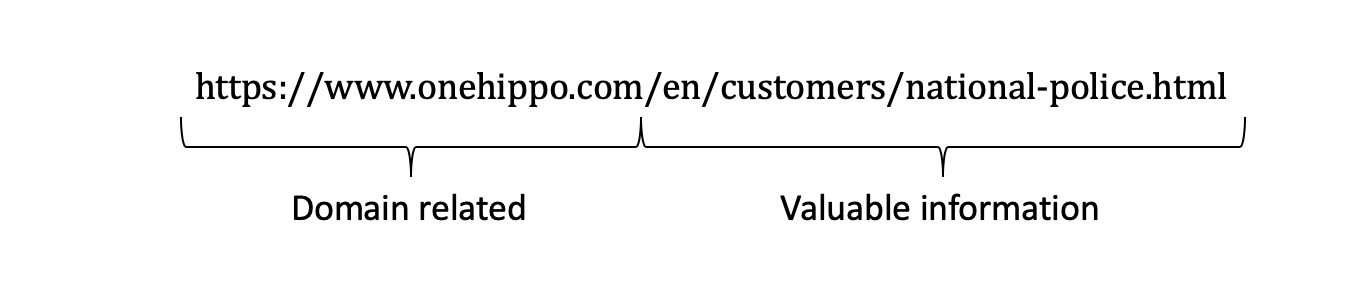


Figure 2 Sample URL

The domain related part is ignored as it brings no value. The valuable information however is filtered via a set of regular expression rules before it is tokenized, stemmed and filtered for stop words. The example from Figure 2 will give us customer, police, national as keywords.

3.1.2  Web scrapping

*Explain why this is needed and what is the motivation behind – such as the fact that nowadays people do not pay that much attention on what the link is about meaning that it might move a bit from the actual content. Discuss the steps taken here.*

Although the initial idea, as explained until now, was to focus primarily and mainly on using only the URL, we decided to add an additional data source in order to compare performance. The specific results can be seen in section 4. Evaluation.

During the creation of the project, multiple meetings were conducted with representatives of the company in order to ensure that the end product will meet the specific requirements. In one of them an interesting hypothesis was put out, that nowadays the importance of the URL has slightly fallen behind. For example, in a google search the majority of people will look into the provided summary of the page, rather than the specific link itself, the content after the domain.

Bearing this in mind we can assume that the URL will not be that helpful when segmenting users, rather the actual content. The hypothesis was changed to a working theory, based on which we developed a framework capable of following a given URL and scrapping the content. As Julia Kho explains web scrapping is a technique to access and extract information from a given website.

Our framework consisted of a so-called spider, the actual scrapper, main repository and two separate repositories. The main repository was straightforward and content the whole text content from the page, excluding any navigational bard, comments and other non-related information. Besides the content, the extracted date was also stored, in the case where the scrapped page did not contain a date identifier. This was done for the sole purpose of having a way to refresh the repository. As this was a side topic in our research, we went for a pretty basic strategy – updating the content in case the available date from the page has changed or update the scrapped content every month or around big system releases. This was done, due to the fact that the majority of the page had to do with different company products, meaning that the content would only change around product updates or releases. The two separate repositories were stored in the form as dictionaries, as follows:

1. {Scrapped URL: keywords from that page}
2. {Scrapped ULR: summary of that page}

The keywords and the summaries were extracted based on ranking algorithm and the available textual content. Important to mention is that there was a separate pre-processing step for the scrapped content. This was needed as the raw scrapped text was filled in with Hyper Text Markup Language (HTML) tags and besides this as some of the pages were documentation related, we had code snippets mixed with text. The pre-processing here was based on a set of regular expression and web-related programming libraries.

The two repositories described above were merged with our user data based on the visited URL’s, thus creating additional features for each user. However, in the process of working we realized that using the summary of a page, will overcomplicate things and does not make sense to use it with the current research in mind. Instead it can be left for Future work. The keywords on the other hand proved more useful as it is show in section 4. Evaluation.

3.2  Methods

*Go over the code; go over research paper again; Make process figure including Neural Network design, back up design decision, IN EVALUATION add Grid Search; Emphasize on steps before the training and the changes, in EVALUATION ADD COMPARISON*

This section will be split into three main parts. The first part will cover the steps taken in order to prepare the data for the Neural Network, whereas the second one will describe the idea behind the model design. The third step will explain the steps taken in order to expand our research. Initially we were only interested in using the keywords extracted from the URL’s, however we decided to investigate whether or not it would make more sense to use the actual content of the URL by scrapping the actual page.

3.2.1  Network Input

*Describe the steps taken in order to prepare the data for the Neural Network*

Following the preprocessing steps described in section 3.1.1 Data pre-processing, we produced a set of keywords per user. However, those keywords were still represented as text and in order for the Neural Network to use it we transformed the text into numeric values using a deep learning library. The initial set of keywords was transformed into a sequence in which each word was replaced by an index value based on word index dictionary. For example, ‘bloomreach’ was replaced by 1, whereas ‘apache’ will be replaced by 228. The dictionary was created based on the frequency of each word in the whole corpus. Lower indexes point to words which are more common and appear more in the given initial corpus.[5] Based on the new sequence the maximal length was taken, which was used to generate a padded sequence for each of the elements of the corpus. Respectively if the given element’s length was lower, than the maximum, 0’s were added.

In parallel, an embedding matrix was initialized. This matrix is based on the word embeddings, vector representation of words learned via Neural Network. The word embeddings are publicly available through the work of Mikolov et al.

The outcome of the following steps will also be discussed in the Evaluation section, as it has to do with testing different techniques than the one used in the initial work of Jiaming Xu et al. – our baseline model.

The next step is to combine the sequence matrix, which was generated earlier, in combination with a weighting factor in order to account for each feature (word) in our sequence. As you will see in the Evaluation section, we ran tests using three approaches as a weighting factor:

* **Binary**: Having the whole corpus, evaluates entries (separate text/document) and returns 1 for each word from the corpus, which is in the given entry. Respectively a 0 is returned if the word is not in the entry.
* **Count**: Following the same logic as the binary approach, it would return 0 if the word is missing, however, if the word appears it will return the number of occurrences. Important here is to account for stop words, such as: the, a etc.
* **Frequency**: 0 will be returned if the given word in not in the processed text/document. If the word appears it will return a proportion of the times the word appears against the total length of the text/document.

The resulting matrix was then normalized in order to ensure that all values have a common scale.[7] Following the normalization, the normalized matrix was combined with the embedding matrix. The last step of the pre-processing is to follow the approach from the baseline method and binarize the product of the two matrixes. As it is based on vector representation, in some cases it would have a negative value, which after the binarization would be changed to 0 and all positive ones will get the value 1 instead.

The following few sentences of this subsection will reflect on what was done in the baseline approach. According to their work, Jiaming Xu et al. are using a binarized (0 or 1) representation of the Average Embedding, in order to train the Neural Network. The binary representation is used instead of labels. In the baseline model they used TF-IDF as a weighting factor, for the features, for the Average Embedding. Term frequency – inverse document frequency (TF-IDF) is used to represent the importance of each word with respect to the its occurrences in the text.[6] Rephrasing, TF assumes that if a word occurs a lot in a given text, then this word should be descriptive for the text. IDF on the other hand, reasons that if a word appears a lot in the given text/document, as well as in others, then most probably this word is not unique for the text and brings meaning. Stop words, for example, which appear a lot in a text bring arguably no context about the content. High score for TF-IDF means that this word is rare and specific for the document at hand.

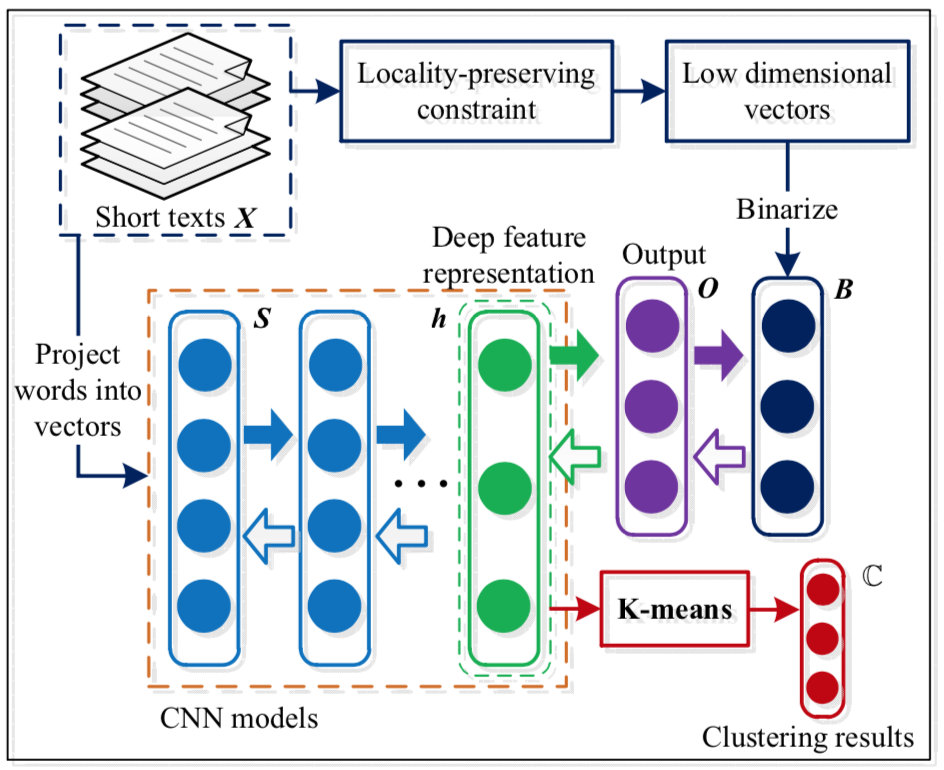


Figure 3 Jiaming Xu et al proposed architecture

3.2.2  Network Design

*Describe the Network design and back up decisions*

This section will cover the design of our Neural Network, together with reasoning for our decisions. Starting from the top – the final design, Figure 4 represent the structure of our current CNN.

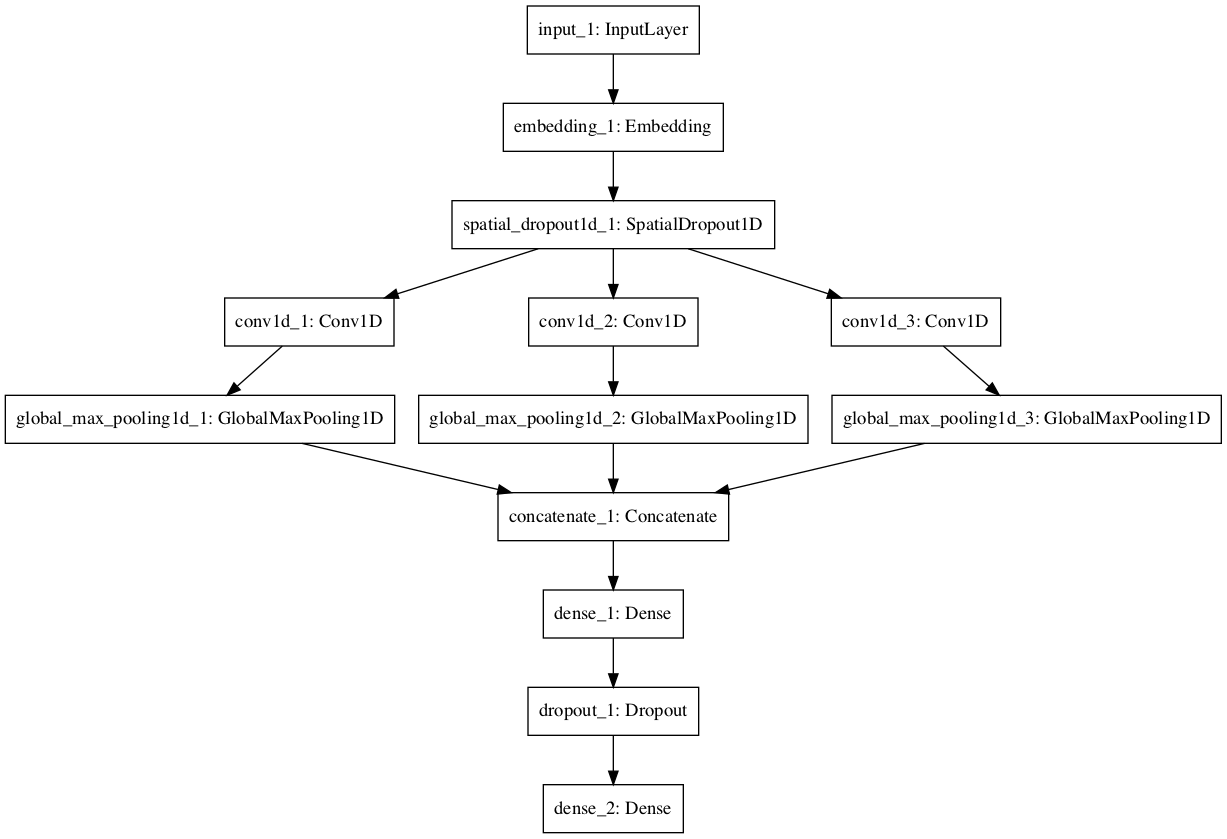


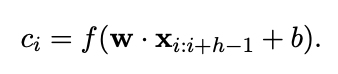
Figure 4 Model Visualization

As we have already discussed we incorporated the same approach used in the works of Jiaming Xu et al. or Yoon Kim, we trained our CNN on top of pretrained word vectors, created and made publicly available by Mikolov et al. The output from section 3.1.1 is used for the training of the network, instead of labels.

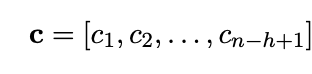
Our neural network starts with an embedding layer, which is a vital part when one is dealing with text in Neural Networks. In our work we are not training our embeddings, rather we simply load the embedding matrix, mentioned in section 3.1.1, as weights. [10] The explanation of Embedding Layer based on the Keras documentation is rather vague [9] and simply states ‘Turns positive integers (indexes) into dense vectors of fixed size’. Jason Brownlee [10] provides a good summary of what embedding’s are and their purpose. Simply put, the embeddings purpose is to represent words as dense vectors, which is based on the representation of the word in a continuous vector space. This approach is a better alternative of one-hot encoding, which has to do with representing each document, from the training examples, as a vector with the size of the vocabulary length, with mostly 0’s. It is on the principle of this word is in the document and this word is not – in this case 0 is applied.

Following different discussions and sources, such as the work of Nitish Srivastava et al on Dropout as a way to prevent Neural Networks of overfitting, we directly apply a Dropout layer to the output of the Embedding Layer. Again, going back to the documentation of Keras we can see that purpose of Dropout is to randomly set a portion of input units to 0 during training. The setting, dropping fraction, is a hyperparameter, which can be tuned. [8] As you can see the from Figure 4 following the Dropout, we have three convolutional layers on the same level.

There are a few things we should be considered in this section of our network design. As numerous sources explain, such as Nils Ackermann in [11] or Jason Brownlee in [12], 2D CNN’s have been used in image processing where the incoming input is of two-dimensional format. 1D CNN’s, however, have been used for other task, such as Natural Language Processing (NLP) related ones, where the input data is of different format. Bearing all this in mind and the task at hand we settled for one dimension. To further build on the use of three layers on the same level, it is important to go back to the idea behind the convolutional layer, which is to simply apply filters or a set of filters, as in our case, to an input. [13] To further build on the use of filters in NLP tasks, we related to the work of Siwei Lai et al. [14]. In the paper, the authors argue that in earlier studies of CNNs in NLP, researchers would rely on filters with fixed sized, however when using such fixed sizes, one is prompt to either loose information, when the size is too small, or have as Siwei Lai et al. point out, an enormous parameter space, when you use larger size. Motivated by both the work of Yoon Kim [16] and Ye Zhang et al [15] we made use of multiple of a set of 3 filters, each with different size. Figure 5 is a shortened version of the work of Ye Zhang [15] and can show you the idea behind the use of multiple filter widths. Basically, each filter will capture a different set of features. As an example, we can look at the work of Yoon Kim [16], where he shows the formula for generating a feature. In that case we have a feature ci coming from a given word window and is given by:



Where f represents an activation function, non-linear, w represents the filter, which is applied to the given window of words and then combined with the bias. This is a single feature, single application of the filter. Once you apply the filter to all words, you end up with a feature map, based on all the window of words.



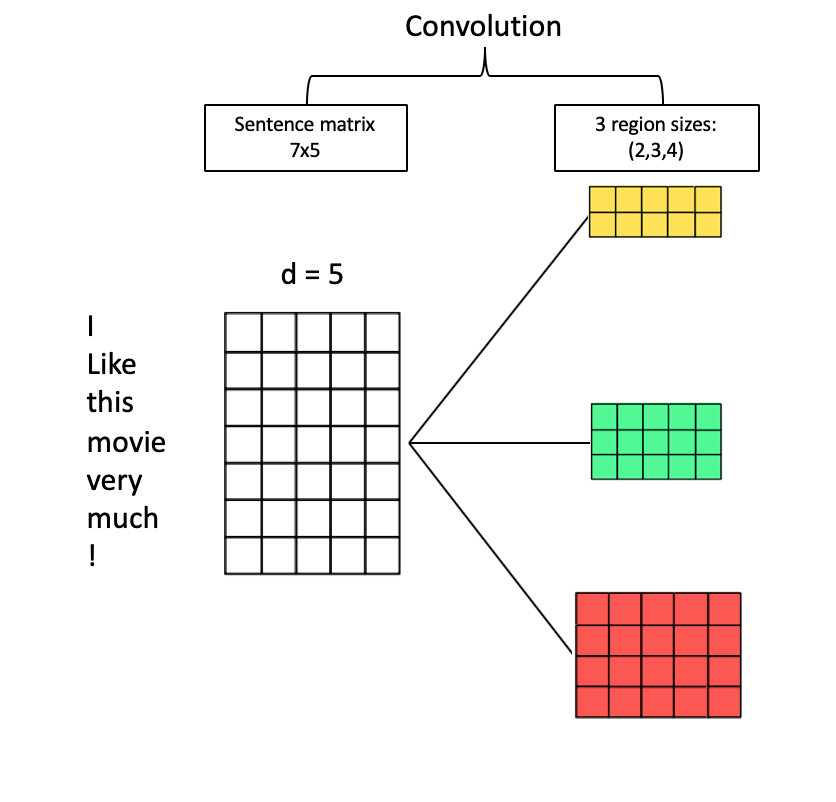


Figure 5 Shortened version based on the work of Ye Zhang [15]

Following the features, we apply the so-called Pooling. As Harsh Pokharna [17] explains, the idea of pooling to reduce the spatial size of the representation and the number of features. The work of Alon Jacovi [18] provides and extensive overview for understanding how CNN are used for Text Classification. Based on this work and other readings we settle for Global Max Pooling. The idea of max pooling is to retrieve the highest value from a feature map.

Following the concatenation of the results we finish the with a combination of fully connected layer, followed by dropout and a dense layer used for the final prediction.

4 Evaluation

*Figures for model performance, comparison between preprocessing, cluster plots, silhouette etc.*

The evaluation of this work will be split in two parts:

* Model comparison

Here we aim at comparing our work to the results presented in the work of Jiaming Xu et al. In their papers the authors are also providing the source of the one of the datasets, which they use. Following the link, we were able to retrieve the same dataset, which is sufficient to test the performance of both models.

* Data inhouse

Although vaguely named, this subsection will account for the comparison of the results versus the results from the research previously done. Both results are based on the same dataset.

5 Conclusion

*Elaborate on research questions & add Future work as a few sentences (Think about it. Maybe expand on scrapping content. Explain the ideas about refreshing the repository from Information Retrieval and why it would make sense to do it – maybe because some pages do not change that much, or you can have a process of updating – assuming your page has edited date, the scrapper can run over weekends and compare edited date from repository with the current edited date).*

The

REFERENCES

*Now they are mixture of Thesis Design and Literature notes + links – Update.*

[1] FNM Surname (2018). Article Title. Journal Title, 10(3), 1–10.

[2] F.N.M Surname, Article Title, https://www.acm.org/publications/proceedings-template.

[3] F.N.M Surname and F.N.M Surname, 2018 Article Title, The title of book two (2nd. ed.). Publisher Name, City, State, Country.

[4] Internet Statistics Facts (Including Mobile) for 2019. 2019. url: https: //hostingfacts.com/internet-facts-stats/.

[5] Keras source code <https://github.com/keras-team/keras-preprocessing/blob/master/keras_preprocessing/text.py>

[6] TF-IDF <https://en.wikipedia.org/wiki/Tf%E2%80%93idf>

[7] Medium Normalization <https://medium.com/@urvashilluniya/why-data-normalization-is-necessary-for-machine-learning-models-681b65a05029>

[8] Keras documentation <https://keras.io/layers/core/>

[9] Embedding layer Keras <https://keras.io/layers/embeddings/>

[10] Embedding Layer example <https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/>

[11] Medium 1D Conv <https://blog.goodaudience.com/introduction-to-1d-convolutional-neural-networks-in-keras-for-time-sequences-3a7ff801a2cf>

[12] MachineLearning 1D Conv <https://machinelearningmastery.com/cnn-models-for-human-activity-recognition-time-series-classification/>

[13] Explanation Conv Layers <https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/>

[14] Recurrent Convolutional Neural Networks for Text Classification

[15] A Sensitivity Analysis of (and Practitioners’ Guide to) Convolutional Neural Networks for Sentence Classification

[16] Convolutional Neural Networks for Sentence Classification

[17] Pooling in CNN <https://medium.com/technologymadeeasy/the-best-explanation-of-convolutional-neural-networks-on-the-internet-fbb8b1ad5df8>

[18] Understanding Convolutional Neural Networks for Text Classification

[19] Max pooling <https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/>

[20] Scrapping <https://towardsdatascience.com/how-to-web-scrape-with-python-in-4-minutes-bc49186a8460>

[21] Digitalization Impact <http://reports.weforum.org/digital-transformation/understanding-the-impact-of-digitalization-on-society/>

[22] Wiki E-commerce <https://en.wikipedia.org/wiki/E-commerce>

[23] Predicting Buyer Behavior <https://becominghuman.ai/predicting-buying-behavior-using-machine-learning-a-case-study-on-sales-prospecting-part-i-3bf455486e5d>

[24] Understanding K-means <https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1>

[25] CRITERIA FOR CHOOSING THE BEST NEURAL NETWORK

[26] An LSTM Approach to Short Text Sentiment Classification with Word Embeddings

Conference Name:ACM Woodstock conference

Conference Short Name:WOODSTOCK’18

Conference Location:El Paso, Texas USA

ISBN:978-1-4503-0000-0/18/06

Year:2018

Date:June

Copyright Year:2018

Copyright Statement:rightsretained

DOI:10.1145/1234567890

RRH: F. Surname et al.

Price:$15.00