Web Mining Search - Project Report - Phase 1

José Monteiro
FCT NOVA
jr.monteiro@campus.fct.unl.pt

Francisco Rodrigues
FCT NOVA
fx.rodrigues@campus.fct.unl.pt

ABSTRACT

News dissemination via social-media is a standard nowadays. The analysis and search of the data from websites like Twitter, could prove fundamental to the creation of news articles with images that are relevant for a certain topic at any given time. We implement and use different feature space searches in order to understand which types can lead to the retrieval of more relevant results according to the query context, that can be either texts or images.

ACM Reference Format:
José Monteiro and Francisco Rodrigues. 2018. Web Mining Search - Project Report - Phase 1. In *Proceedings of Web Mining Search (Web Mining Search - Project - Phase 1).* , 4 pages.

1 INTRODUCTION

This project's goal is to, given a news story topic and a stream of social-media text and images, illustrate a news story with social-media content by linking a story-segment to an image or text relevant to it. The social-media data is collected from Twitter and consists of over 2k images with additional metadata regarding the tweet, each associated to the Edfest2016 event.

The first phase of the implementation consists in creating a search method to retrieve the most similar data to a visual or textual query, by setting up the test bed with search spaces for each individual feature space, namely: Bag of Words, Bag of Entities, Histogram of Colors, Histogram of Gradients and a Convolutional Neural Network. The queries are read from news topics provided in JSON format and each topic consists of three to four news segments containing a segment id, text, image and keywords. In this phase, only the segment's text or image is used to retrieve the most similar documents according to a given feature. Then an observational analysis is made where the the best matches from each individual search space are observed to discuss the following issues: are all the retrieved documents relevant to the query topic? Which feature spaces performed better and why?

2 ANALYSIS OF TWITTER TEXTS

The increasing adoption of platforms like Twitter to express views and opinions, made social media an exotic medium for data scientists to research. As with all data research, a basic understanding of the underlying data set is needed to achieve better results. This section consists in the study that was made to complement the techniques that were lectured in the classes in order to develop this work.

2.1 Content and Syntax

The information inside a tweet differs from the usual information that we find in a blog post or a news article. Whereas in the last two we usually find texts containing a lot of useful information about a certain topic without spelling mistakes or abbreviations, in the Twitter world tweets are limited to only 140 characters so one can frequently find in their content poorly structured sentences, abbreviations and slang words that are not found in the dictionary [2].

The presence of noise in the tweet content presents itself as an obstacle to those who, like us, are performing a task of tweets analysis for text based search. Special characters, URL links, hashtags, user handles or words with repeated letters need to be processed in order to extract relevant information from the tweet.

Hashtags and handles provide important information about the author and/or persons involved and the topic of the content respectively. So when we talk about removing them, we are actually referring to the removal of the special chars '' and '@' and not the whole text.

2.2 Semantics

Usually when trying to find similar documents in a given data set, some tokenization process is performed over every element in the set. This can be: filtering some characters, removing duplicates, or using only words that appear more then a certain number of times in all the documents and producing tokens. Tokens are words, that are a representation of the original sentence. One of the most used filtering methods that is applied together with other filters is the Stop Words filter, which eliminates from the vocabulary all the stop words of a given language.

Stop words carry the sentiment of a sentence, and make it possible for Named Entity Recognition (NER) algorithms to identify the dependence between words to establish which of these are entities. For that reason we choose to filter or not filter stop words depending on the type of feature space we are using. Note that, because of size restrictions and the presence of abbreviations [4], like we previously referred, in the majority of the tweets, stop words and connection terms are simply not present. This happens because the users choose to omit them in order to keep the character count bellow the maximum allowed limit, which can can largely difficult the identification of entities when using an NER algorithm [3].

3 IMPLEMENTATION

In this section, it is described the implementation of each feature search space, as well as the pre-processing and query preparation approaches that were taken to read the query JSON file and run the tests.

3.1 Pre-processing and Tests Framework

The pandas library is used to read data set csv file. In this phase, only the tweet's text and image-url fields are used, which are stored in two different arrays and the data read from the image-url field is

edited so each image points to the folder containing the respective file.

Since all the search spaces require the images to have the same size to prevent unexpected results, all images are center cropped to 224*224, preserving the aspect ratio. The same cropping process occurs when an image is provided as a query.

The JSON file containing the story topics is read with the "json" library and the test framework consists in permutating , for each story and segments, the most relevant parameters of each search space. The output of each search space are the top 5 results of the search, with the id of the tweet and the distance to the original query under the applied parameters of that test.

3.2 Bag of Words

The Bag of Words search space is implemented as lectured in the lab lessons. An external tokenizer is used, called Tweet Tokenizer. It can replace tweet features like usernames, urls, phone numbers, times, etc. with tokens in order to reduce feature set complexity and improve performance. This tokenizer is tested against the Treebank Word Tokenizer. The other relevant parameters that are being tested are the following:

- **Lemmatization** - Tests are performed with and without lemmatization.
- **Distance Metrics** - Cosine and Jaccard metrics are tested for textual search queries.

The vectorizer is set to use the "english" stop word list and to convert all characters to lowercase before tokenizing. The minimum document frequency is set to 3 in order to reduce vocabulary size.

3.3 Bag of Entities

The Bag of Entities search space is implemented as lectured in the lab lessons. The additional steps needed to achieve a Bag of Entities are similar to the Bag of Words, but in this case, after extracting the entities from the original query, the vocabulary will be the entities extracted from the twitter texts. For this test, the tweet tokenizer is compared against the Treebank word tokanizer. The minimum document frequency is set to 3 in order to reduce vocabulary size.

3.4 Histogram of colors

The Histogram of Colors search space is implemented as lectured in the lab lessons. In this search space, the following parameters are tested:

- **Distance Metrics** - The distance metrics tested in this search space are euclidean, cityblock/manhattan, minkowski, cosine.
- **HSV and RGB color spaces** - Both models are tested.

3.5 Histogram of Gradients

The Histogram of Gradients search space is implemented as lectured in the lab lessons. In this search space, the following parameters are tested:

- **Orientation** - The number of orientations tested are 8, 9 and 15.
- **Pixels per cell** - The pixels per cell parameter is tested for 8, 16 and 32.
- **Distance Metrics** - The distance metrics tested in this search space are euclidean, cityblock/manhattan, minkowski, cosine.

3.6 Convolutional Neural Network VGG16

The Convolutional Neural Network search space is implemented as lectured in the lab lessons. The additional steps taken to achieve the desired architecture are the inclusion of the VGG16 and, after extracting the features of the VGG and predicting the image concepts of the query image and dataset, a k-neighbours approach is made to get the 5 most probably related images.

4 RESULT ANALYSIS

In this section, the results obtained from the tests are discussed to understand why some search spaces perform better than others under determined circumstances. Be it the parameters used when performing the tests, or the nature of the query itself.

4.1 Tokenizers

Using the 3rd party tweet aware tokenizer produces better results because sometimes information in handles and tags is important. Also the trimming of links is a useful way to reduce the vocabulary. The default Word Tokenizer can't efficiently tokenize some of the tweet text and leads to useless terms that 99% of the time don't qualify as entities (in the BoE approach) even though there is information in the tweet that is for sure an entity. It also tokenizes links which can lead to similarities that are not relevant and misdirect the search. ex: pic.twitter.com as a token, every tweet has one so it will misdirect the vector distance calculation making it think that the tweets share some useful info.

4.2 Bag of Words

The analysis work we did a-priori regarding the content and semantics of tweets was very useful in order to understand what would be necessary to extract meaningful words from the query texts. Realizing also that abbreviations could be present and making sure to somehow represent them in their primitive form were the challenges for this feature space.

In our implementation, tests of the BoW the parameters alternated were the lemmatization (True or False) and the distance function (Cosine or Jaccard).

By lemmatization we mean morphological analysis of each word and return it to it's base form. With this we would expect to find in our results texts which could be using different words that mean the same and have the same base form in order to find texts that are similar but were written in a different way.

The Jaccard Similarity is defined as size of intersection divided by size of union of two sets. Since sets are used, if a word appeared twice in a sentence it will count as only one occurrence. This does not happen with Cosine Similarity, since it calculates similarity by measuring the cosine angle between two vectors, thus the need to convert sentences into vectors as in this work's Bag of Words approach, in which a word frequency is performed. Therefore, the presence of a repeated word will change the Cosine but not the Jaccard similarity. The result is that Jaccard similarity is well applied when duplication does not matter, and Cosine similarity is well

applied in cases where duplication matters when dealing with text similarity.

However, in the performed tests, neither metric proved to be better than the other and the results were identical. In contrast, sometimes lemmatization included in the top 10 results some relevant documents that were absent without it's use, while preserving the rest of the results. But in some queries, the use of lemmatization excluded from the top relevant documents, independently of the metric applied (Jaccard or Cosine).

In terms of the images associated with the top documents found, most of them are representative of the content of the tweet but that is not always the case, because the text is completely independent of the image content of the tweets. Even though the intended use of attaching a picture to a tweet text is to complement it, this is not always true, it is up to the person what the picture attached to the tweet will be and vice versa.

4.3 Bag of Entities

Using the NLP tools provided by the NLTK library, the default sentence segmenter and the parts of speech tagging together with a third party tweet aware tokenizer (used in 4.2), the results for queries and tweets which contain long texts with context and are properly capitalized, i.e semantic relations exists, are satisfactory and most of the time relevant to the topic that is being searched. This happens because it relies on entities instead of words by themselves. Because of this, the same type of results don't happen when either the query text or the tweets text is short and lacks the necessary terms for the NLP to extract the entities used for a bag of words type search. This comes to no surprise, and it has been the topic of a research by the computer science community [3], the main cause being that the NLP models are trained to tag words in news texts which are more structured and contain more characters/words, this means that the performance of the tagging drops when applied to tweets, which characteristics are very different from news articles.

Solving these problems could be done by using a parts-of-speech tagger that is prepared to receive tweets as input, like the one proposed in *Part-of-speech Tagging for Twitter: Annotation, Features, and Experiments*[1], and the full code is available for public use.

4.4 Histogram of Colors / Gradients

In both feature spaces we do the same image pre-processing of center cropping the image to a square size, and the results for each are very similar in terms of similarity and relevance.

The histogram of Colors results present us with images that are similar in color to the query image, while the histogram of gradients produces results that have similar shapes when compared to the query image.

For the histogram of colors, the tests are conducted with bins of 4 and 8 each space of the HSV model, and the calculus of the similarity is performed with pairwise distance using the Euclidian, Cityblock, Minkowski and Cosine. The results show that with a higher number of bins the search space performed poorly and started to show dissimilar objects in the top results, while with fewer number of bins the top results showed similar images in colors for most of the segment's top results, leading us to believe that bins between 4 and 8 are adequate to score similar objects in this data set. The metric applied didn't present drastic changes other than changing the order of some of the top results. The same can be said when testing for the RGB model, even tough the results were sometimes different, the conclusions remained the same.

For the histogram of gradients search space, test are conducted to test the number of orientations, pixels per cell and distance metrics. The results didn't vary between 8, 9 and 15 orientations, maybe due to the fact that this values are almost the same. The the same was observed for each metric used and the main difference was the order displayed by some top results, but not the set of images itself. However, much better results were obtained when using 32 and 16 pixels per cell than with 8, displaying some similar and identical images in the output. This search space proved to be good when trying to detect shapes in the images and finding duplicates, but the value of the parameters must be optimized for a specific problem to achieve better results.

Similarity between query documents and the top retrieve documents is very high, but it is not what we were trying to achieve with this project, where the main goal is to find documents that have some connection to the query topic/context. We can conclude then that almost not all documents are relevant, we say not all because if there are images that are near identical to the query image they will show up in the documents retrieved, but other then that, there is no relevant connection at all.

4.5 Convolutional Neural Network (VGG16)

The performance of this model in the image based search is directly linked to how good are the predictions made by it. This predictions in turn can be affected by image rotation, re-scaling and color. Given this and the fact that we center crop every image in the pre-processing stage, the tagging performance is from the start crippled, which can lead to incorrect tagging and can mislead the search.

Even knowing this, we knew the center crop process had to be done in order to represent the images in the same size to help get a more uniform data set representation. The results show us that CNN with VGG16 [5] is the best way to find images with an image query, when compared to the HoC and HoG feature spaces, and that the tagging process is optimal when the concepts of the query image match the 1000 concepts which the model can accurately predict. For example, in the news segment that talked about food and the images of the segments only had food present, the top 5 results showed documents with images that contain the exact same type of food or a very similar one. This didn't happen for query images that contained a lot of object information and confused the model.

The VGG16 [5] model, although it performs reasonably well, it is no longer state-of-the-art so there are other models like the Res50Net which can do a better job of tagging images in a much shorter time. Nonetheless VGG16 has useful features like giving the possibility to explore the concepts of intermediate layers and doesn't require the optimization of parameters like in the HoG and HoC, but the trade off is speed in the execution.

5 DISCUSSION

Relevance comes in regards to topic context, it means finding documents that contain in their text or image some connection to the query topic. Similarity is the measure of how much alike two documents are. Similarity measure in the context of Web Mining and Search is a distance with dimensions representing features of the objects. If this distance is small, it will be the high degree of similarity where large distance will be the low degree of similarity.

Having established the difference between this terms we come to the conclusion that, according to the result analysis of 4, for text based queries the best results come with the application of the Bag of Words approach. Using images as the query, the CNN method presents the more relevant results with a trade off in execution time, where the HoC and HoG perform better if the parameters are optimized for a specific object, like finding cars in images.

Closing the gap between similarity and relevance is a matter of joining different types results in order to approximate them from the topic context. When searching using the feature space of 4.4 we get results that are similar to the image query of the segment, but almost not all of them are relevant, but if we also use the top result's associated tweet content we can match it using the feature space of 4.2 and approximate the retrieved documents to the query topic.

Another approach to mitigate the gap can also be retrieving images with text. Searching for images with a text query instead of an image like we used in 4.5, we can extract the concepts identified for each image (tags), and use them in a bag of words manner to compare with the text query. The bag of entities method explained in 4.3 could also be used to narrow even more the concepts, but as we saw in the result analysis the recognition of concepts using the NLTK library didn't prove to be very efficient, so the results could not be the best. It would ultimately depend on how good would the query text be in terms of semantics.

REFERENCES

[1] Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-speech Tagging for Twitter: Annotation, Features, and Experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2 (HLT '11)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 42–47. http://dl.acm.org/citation.cfm?id=2002736.2002747

[2] Ozer Ozdikis, Pinar Senkul, and Halit Oguztuzun. 2014. *Context Based Semantic Relations in Tweets.* Springer International Publishing, Cham, 35–52. https://doi.org/10.1007/978-3-319-05912-9_2

[3] Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named Entity Recognition in Tweets: An Experimental Study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '11)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 1524–1534. http://dl.acm.org/citation.cfm?id=2145432.2145595

[4] Hassan Saif, Yulan He, Miriam Fernandez, and Harith Alani. 2014. Semantic Patterns for Sentiment Analysis of Twitter. (10 2014).

[5] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014). arXiv:1409.1556 http://arxiv.org/abs/1409.1556