Cluster City

Text & Multimedia Mining Project

Dennis Dorrich (s1001070)

January 17, 2018

Abstract

Cluster City is an approach to group cities into k different clusters, based solely on their description text, using k-means clustering. City descriptions were obtained from wikivoyage.org, which is an extensive online travel guide that contains text descriptions about different kind of holiday destinations and cities. Feature extraction contained among others BoW, word-, pos-tag- and named entity count. The so gained clusters were tested on human labelling in order to assess whether their underlying structure corresponds to human heuristics. In applications the clusters can be used for recommending cities, based on previously interested cities. It can also be used to map free text-input, such as *'My city is a capital and has a precious ancient centre'*, to either of the clusters and recommend similar destinations.

Contents

1 Introduction **3**

2 Methods **4**

2.1 Text Corpus 4

2.2 Pipeline 5

2.3 Feature Extraction 5

2.4 Unsupervised Clustering 5

2.5 Human Labelling Task 6

2.6 Implementation 7

3 Results **7**

4 Analysis **8**

4.1 2 Clusters 9

4.2 3 Clusters 10

4.3 4 Clusters 11

5 Discussion **11**

1 Introduction

Travel websites like tui.com or expedia.com, which are commonly used to select a holiday destination are very bound to their catalogue and selected offers when it is about finding a novel travel destination. Other websites, that are more adaptive to personal preferences, like http://www.jauntaroo.com let the users select preferred weather, things they'd like to do, ambiente etc. Where these approaches already offer more flexibility, they are still bound to fixed, pre-defined categories. Within their catalogue, each destination had to be hand-assigned to the respective features. Also the query needs to be in an exact specified structure in order to match it against the candidates. More elaborate recommender systems like on http://www.amazon.com try to find associations (Adomavicius and Tuzhilin (2005)), like if product A was purchased then product B was bought probably as well. This is an interesting approach, as it doesn't depend on inherent product details but only on co-occurrences between market basket items. This paper goes one step further and provides a way to group similar items together, independent of user-data and of hand-crafted features: From the platform wikivoyage.org, that describes itself as an online travel-guide, raw text data of city-descriptions was retrieved, for example:

Rome (Italian: Roma), the 'Eternal City', is the capital and largest city of Italy...

Based on that, after feature extraction (see section Methods), unsupervised clustering was performed to group similar cities together. When we think about the given task, putting cities into boxes, then most of us would think about categories like "same country" or "capital vs. no capital". The Cluster City approach could reveal new structures that we normally wouldn't think of and could consequently change the way how travel websites and catalogues are structured.

Another practical application would be to allow free text input like "My city should be on the beach, be a capital and people should speak Spanish", which can then be mapped to one of the clusters by the pre-trained algorithm and return destinations that are similar (within the same cluster).

A crucial question before using the algorithm for the applications above is to know how well it actually performs. Intrinsic measures of cluster validity like cohesion and separation (Tan et al. (2013), Wang et al. (2009)), are useful to evaluate quantitatively whether for instance 3 or 4 clusters perform better, but it doesn't give an insight whether the clustering matches the underlying structure, that humans would perceive. An example of this is visualized in figure (Figure 1). This is why external information was used in order to evaluate the quality of the gained clusters. Usually with external information the true indices are meant. In our case true indices would mean we already knew the wanted structure, which is exactly what we want to avoid. Instead of this we let fellow students and friends assess to which extend the clusters give intuitive sense for humans. More about that in section 2.

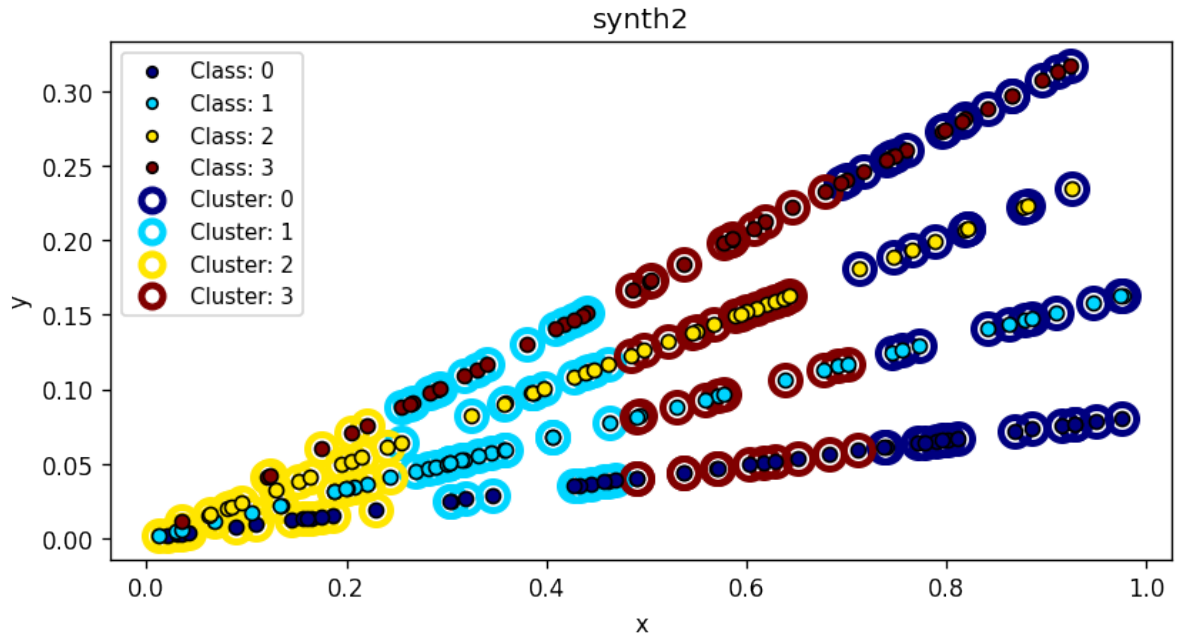

Figure 1: Shown are datapoints from an example data-set. The underlying structure is clearly 4 linear functions of first degree, as shown by the dot-colors that represant the true labels. The k-means clustering algorithm however finds 4 cloud-structures, shown by the colored circles around the dots.

The **research question** is: How well does unsupervised City Clustering result clusters whose structure is understandable for humans. The null-hypothesis in this example would be that there is no structure recognizable by humans.

In the next sections we will first go more in detail about the methods (section 2), especially for the pipeline and feature-extraction. Also the experiment with the human labelling task(subsection 2.5) will be described in depth and visually. After that the results are presented and analyzed in order to answer the research question, proposed above.

2 Methods

2.1 Text Corpus

Wikivoyage offers to download their data-set in English, containing the xml text for all articles, which is 11435 travel destinations. Almost all articles are structured in the same manner, such as *Short summary*, *Get around*, *See*, *Things to Do*, etc. For the initial version we took only the short summary, as all city descriptions had this one in common.

2.2 Pipeline

Starting with the database dump from subsection 2.1, the whole pipeline existed of the steps in the enumeration below. Feature Generation, Clustering and the human labelling task are explained more in detail below in their respective section.

- Preprocessing
    1. Clean raw text from xml tags, code for pictures etc. and removing stop-words.
    2. Split Each city description into different paragraphs, like "Short Summary", "How to get there", or "History". This is still a simple aggregation step, as the layout for each city is relatively similar, following the same order.
- Feature Generation and Extraction: Using e.g Regular Expressions, word-embeddings, or other predefined features like word-count or named-entity-recognition.
- Perform unsupervised clustering on features
- Human Labelling Task

2.3 Feature Extraction

After cleaning the raw-text, features has been extracted by the following methods. With a Bag-of-words approach, we trained a count-vectorizer on the whole corpus in order to learn the vocabulary of the 5000 most frequent words (after removing stopwords, given by the nltk library). By this, each text can be converted to a 5000-dimensional vector, where each entry counts how often the respective word appears in the text.

Next to that, we used again the nltk library to get for each word its position tag, such as e.g *VBD* for *Verb, past tense*. We then simply counted, how often each pos-tag appeared.

Analogue to that, it was also counted, how many words the text contained and how many named entities were detected by nltk.

Based on these feature representations for each destination, we can now apply the clustering algorithm.

2.4 Unsupervised Clustering

For grouping the cities together by their similarity in the feature space, k-means clustering was used (Hartigan and Wong (1979)). Bearing in mind that also the clusters had to be assessed by humans later, k was in the range of 2 to 4. This was a consequence of preliminary discussions with first participants who reported that already with 4 clusters it becomes hard to create a distinct personal image or representation for each of them.

2.5 Human Labelling Task

In order to test whether the clusters give intuitively sense for humans, fellow students and friends, in total 34 participants, were presented with the following task.

In each question they are presented with 2-4 word-clouds, from which each contains 5 cities, plus one yet unlabelled city which has to be assigned most sensibly to one of the clouds (see Figure 2). This procedure of assigning a new city to one out of 2-4 clusters is repeated 5 times for 2, 3 and 4 clusters each.

Each of the clouds is a subset of the generated clusters. These subsets were selected by 2 conditions: First, I allowed only the 100 most visited cities in the world EuroMonitor (2017), because otherwise, by picking totally random we would get results like for instance *Rurrenabaque,Nha Trang,Oberlin,Kala* for cluster representatives, which are most likely unknown by most participants. From these 100 cities I reduced the set to 71 cities, because Asian and especially Chinese cities were highly over-represented due to their intra-continental tourism and less known in Europe. Secondly for each cluster I took the 5 destinations that were closest to the cluster centroid in feature space. The number 5 results from a similar consideration as above with the number of clusters. 5 words per cloud were perceived as manageable, whereas 10 would be too much input.

The city that needs to be assigned is than accordingly the 6th closest to the centre. This means, we know how our algorithm assigned the city and now we test if a) participants would follow the same logic and b) agree among each other.

The new cities vary, whereas the clusters stay the same content-wise. Although the word-clouds had to contain the same cities, they were generated randomly for each task, so that colour and position could be ruled out as a systematic bias.

The assignment is done solely by the participants intuition, without having to elaborate on their idea of the cluster.

This means that In the experiment the clusters do not have to be given explicit names like “popular” and “unpopular” in order to contain a structure that makes them valuable for applications.

Figure 2: The figure shows an example of the survey where participants had to assign a city to one of the presented clusters.

2.6 Implementation

The pipeline was completely programmed in Python 2. For Preprocessing it was made use of the regular expressions library. Feature Extraction relied on the nltk-library (Manning et al. (2014)) for pos-tags and named-entity recognition. The scikit-learn machine learning libray helped for easier using count-vectorizer and mainly for kmeans-clustering. The word-clouds were generated by using the python library "wordcloud". The survey was carried out by a Google form.

3 Results

Below in Figure 3, Figure 4 and Figure 5 the results of the survey are shown. The percentages are gained by simply counting how often which cluster was voted and then dividing by the total number of votes.

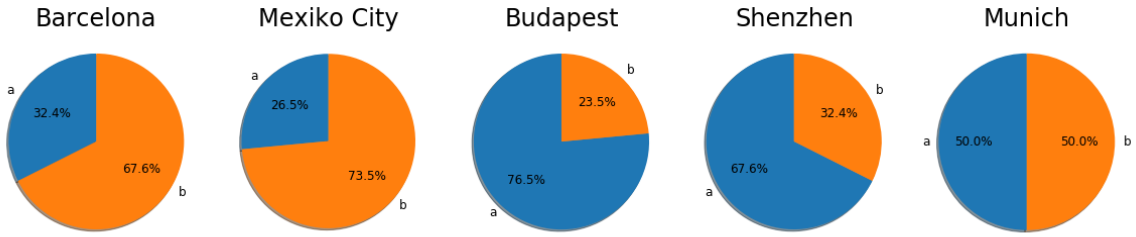

Figure 3: Shown are the results for the city assignments of 2 clusters. Blue is showing the percentage of cluster a and orange of cluster b.

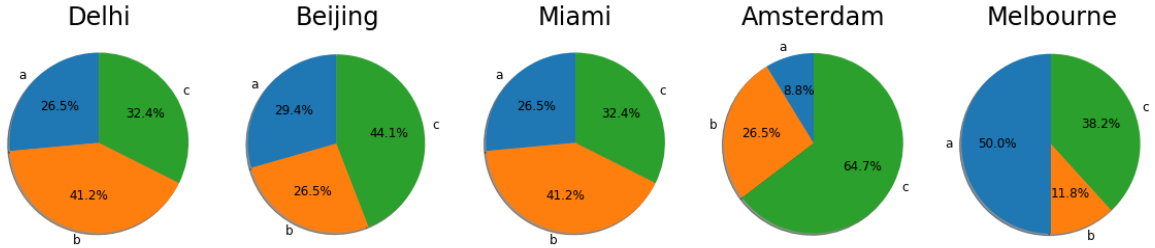

Figure 4: Shown are the results for the city assignments of 3 clusters. Blue is showing the percentage of cluster a, orange of cluster b and green of cluster c.

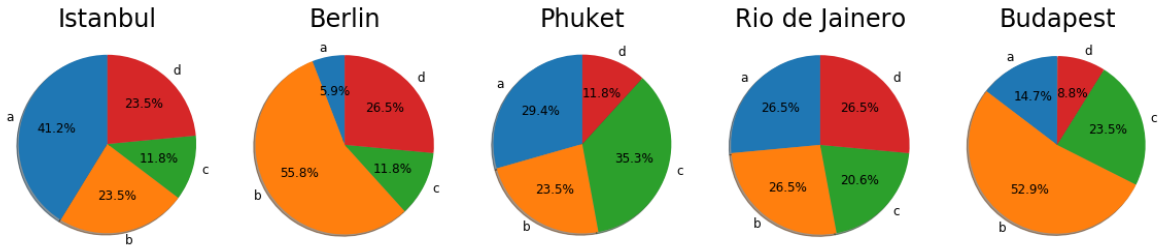

Figure 5: Shown are the results for the city assignments of 4 clusters. Blue is showing the percentage of cluster a, orange of cluster b, green cluster c and red cluster d.

From the results we can already see, that for some city certain clusters were preferred. In the analysis we test this relationship on statistical significance and compare it to the *true clusters*.

4 Analysis

Supposing that the answers were given randomly, we would expect that for 2 clusters 50% chose cluster a and 50% cluster b. The same thing applies for 33 and 25% for 3 and 4 cluster accordingly. With a one-tailed binomial test we can see exactly if k out of n=34

drawings, given the null-hypothesis above, is statistically significant. The threshold of significance was 0.05.

4.1 2 Clusters

Figure 6: Shown are the 2 clusters the participants had to assign cities to. The following cities belonged to a and b: a) Mexiko-City, Shenzhen b) Barcelona, Budapest, Munich

For 2 clusters, Budapest and Shenzen were significantly often assigned to cluster a and Mexiko and Barcelona were significantly often assigned to cluster b. For Munich there was no preference.

Although in all cases except of Munich the raters show relevant preference to one cluster, this is only in 50% of the cases the *true* cluster from the k-means clustering.

4.2 3 Clusters

Figure 7: Shown are the 3 clusters the participants had to assign cities to. The following cities belonged to a, b and c: a) Delhi, Miami, Melbourne b) c) Beijing, Amsterdam

For 3 Clusters, we didn't find any significance values for Beijing, Delhi and Miami. For Amsterdam, participants agreed that it is not cluster c, but a.[1] For Melbourne the preference was that it's not b, but a.

The only correctly assigned city was Melbourne.

[1] *not cluster x* means, that significantly few participants voted for cluster x. Which means, they can still be unsure which cluster it is, but sure about which to discard.

4.3 4 Clusters

Figure 8: Shown are the 4 clusters the participants had to assign cities to. The following cities belonged to a, b, c and d: a) Phuket b) Istanbul, Rio de Janeiro, Budapest c) d) Berlin

- Istanbul was thought to be not to c, but a.: Wrong
- Berlin was assigned not to a, not to c, but b.: Wrong
- Phuket was assigned to not be in d: Correct
- and Budapest to not be in a or d, but b: Correct
- Only for Rio de Janeiro there was no significance found.

Again we do find significant agreements among the raters, but only in 50% of the cases they go along with the correct cluster.

5 Discussion

From the results after acquiring 34 participants we can say, that in many cases participants agree on where to assign the cities. This preference however was not necessarily in accord with the *true* cluster assignments that resulted from k-means clustering. In about 50% of the cases, the assignment by humans and by k-means differed. Hence we can conclude, that humans did in many cases agree on some underlying structure, but this structure is not the same as represented in the feature space, generated from the text descriptions.

We see 2 main reasons that explain the results. First, the text-data per city, that we used to train the model, only existed of a few sentences in average. As these sentences come from the *short description*, they share often the same structure, e.g *A is a city in B. It was founded in the year C, by D*, where A-D are replaced with the respective value. This similarity also leads to a similarity in the feature-space and makes the clustering harder or less indicative. By using longer texts, that contain more information, the clustering could be improved. Secondly, in this first study the feature extraction was still rather basic. Because of time-constraints it was limited to mostly counting appearances of different attributes. By this use of a count-vectorizer, the sequential effect of text-data is completely lost. A first way to improve would be to extend Bag-of-words to n-grams, not only for the vocabulary, but also for position-tags. Then, with even more time, using for instance a LSTM-network would enable to have text of variable input-size and still maintain the sequential information. Also word-embeddings like word2vec (Mikolov et al. (2013)) that project a word into a high-dimensional vector space could be used additionally, for instance as input for the LSTM-network. Another severe limitation that we found, especially for cities where known sights and buildings play a key-role, is to only count how many named-entities appeared in the text. The most logical extension here would be to create a vocabulary, just like in the BoW-approach, but only with named-entities.

Next to the feature-extraction, the experiment was designed by decisions according to the feedback, coming from the first round of participants. The number of clusters and number of words per clusters were selected by only qualitative feedback of the 5 first test object. These values however are not carved in stone and can be varied in future experiments in order to see their effect on the outcome. After the experiment we got oral feedback from most of the participants. Although most tried to answer just by intuition, they reported independently of each other that the attributes *geography*, *language*, *religion* and *economy* governed most their allocation. Thinking about the text-descriptions, these attributes are at the most implicitly encoded in the feature-representation. Another limitation in the experiment that was made to keep the survey feasible for the participants, was to keep the clusters always the same. This could have introduced a systematic effect of having exactly these clusters.

In total, we have seen that the framework of this study was compromised to meet the given time-limitation. This might have introduced some systematic biases. In order to exclude these biases, the design can easily be extended, with the suggestions above, in order to conduct new and possibly more extensive studies on the given subject.

The conclusion above, that humans' reasoning on how to put cities most reasonably into boxes differs from how the algorithm does it, doesn't exclude the possibility that *Cluster City* might still be useful for e.g a recommender system. Humans don't have to understand why exactly these new travel-destinations or products are recommended in order to be good recommendations. In this case, its validity still has to be proven otherwise, for example by looking at how it impacts the sales rate. The study also delivers a framework, in which free text-input[2] can be mapped to the most similar city, given by its text-description.

[2]With free-text-input any text of undefined length is meant. This can be a city-description from the data-set or a personally compiled one.

References

Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749.

EuroMonitor (2017). Internationals top city destinations ranking. Retrieved: 2017-12-23. `https://blog.euromonitor.com\/2014/01/euromonitor-internationals-top-city-destinati` `html`.

Hartigan, J. A. and Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., and McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tan, P.-N., Steinbach, M., and Kumar, V. (2013). Data mining cluster analysis: basic concepts and algorithms. *Introduction to data mining*.

Wang, K., Wang, B., and Peng, L. (2009). Cvap: validation for cluster analyses. *Data Science Journal*, 8:88–93.