Using clustering to filter results of an Information Retrieval system

Pilar López-Úbeda and Manuel Carlos Díaz-Galiano and
Maria-Teresa Martín-Valdivia and L. Alfonso Ureña-López

Department of Computer Science, Advanced Studies Center in ICT (CEATIC)
Universidad de Jaén, Campus Las Lagunillas, 23071, Jaén, Spain
{plubeda, mcdiaz, maite, laurena}@ujaen.es

Abstract

In this paper we present our participation as SINAI research group from the Universidad de Jaén at Text REtrieval Conceference (TREC) in the News task. Specifically we have participated in sub-task 1 called Background Linking. In this task we try to apply K-means clustering algorithms to obtain related news from different domains and topics. We also use document reordering techniques to obtain a new ordered list of relevant articles. For text processing we use the popular TF-IDF technique. The results obtained have not overcome the proposed baseline, although we are usually above average, improving in some cases 78% the average.

1 Introduction

Nowadays, we have access to huge digital information. For this reason we need an efficient search process to obtain relevant information to a specific query. Taking into account that there is a greater number of documents to process and that we need greater precision and accuracy in the results obtained, the information retrieval process is everyday more costly and difficult.

Specifically, the news industry has changed rapidly in recent years, as most people, and especially young people, receive information digitally. According to a study by Pew Research (Mitchell et al., 2016) , approximately 38% of Americans receive information digitally.

Information Retrieval Systems (IRS) play an important role in the process of efficient searches, offering relevant documents to a user, these documents are obtained from a previously processed collection. The IRS pays attention to different tasks: storing documents, evaluation of retrieved documents, ranking of documents and presentation of results.

The News tracks (Huang et al., 2018) in the Text REtrieval Conference (TREC) uses advanced searches to put in context the reader of a news of the Washington Post obtaining other related news. The task is divided into two subtasks. Our group has participated in subtask 1: Background Linking.

The goal of Background Linking task is to develop systems that can help users contextualize news articles as they are reading them. Given a specific article we need to recommend other articles that this person should read below that are most useful for providing context and background.

The rest of the paper is organized as follows: Section 2 describes the data collection, the system is described in Section 3; Section 4 contains the results obtained, and finally, sample conclusions are shown in Section 5.

2 Data collection

The data for this task have been taken from the Washington Post newspaper (NIST, 2018) and include five years of published articles from 2012 to 2017. The collection consists of 595,037 articles.

The articles are in JSON format and the content is divided into paragraphs. It is possible that the article contains extra information from images, videos, tweet or instagram. The article also contains a URL pointing to the Washington Post website.

2.1 Collection processing

The collection has been processed and each article has been prepared for indexing.

To create the index we have taken into account the following common fields in all the articles: *id, type, title, article_url* and *published_date*. And different types of content of

the articles: *ar-wikitude, author_info, deck, image, inline_story, inline_story, kicker, list, pull_quote, sanitized_html, top_deck, tweet* and *video*.

The last step was to remove the found HTML tags of all the indexed fields.

2.2 Index

In our case, we index information about the terms in the document collection, which it can be accessed later using either a term or a document as the reference.

For this task we use the Indri search engine. Indri is a component of the Lemur toolkit (Croft and Callan, 2016) for Information Retrieval. The Lemur toolkit is an open-source system built to enable Information Retrieval research using language modeling. Indri extends upon the goals of the Lemur toolkit by adding new functionality for the retrieval of semi-structured documents and a flexible query language.

3 Methodology

In this section, we will describe the strategy followed for the task. Our system tries to group documents by categories using the K-means clustering algorithm applied to document classification and processing.

The architecture followed to develop our system is shown in Figure 1 and below we will describe each step.

- **Extract document by DocId**

  The first step in this system is to take the DocId of each topic and make a dump on the collection of documents. With the DocId we obtain each complete article.

  For this task we used different parts of the article: the title, the abstract and the title together with the abstract. We use these parts because we consider them to be the most informative parts of the news. The title contains the content of the article in abbreviated form, and with the abstract we obtain the complete content of the news.

- **Create query**

  According to the parts we get from the article we prepare different runs. With each part of the article we build the query that will be passed to the Information Retrieval System (IRS).

  In order to correctly treat each query by the IRS, we have normalized the text included in each one removing hyphens that are not between words, special characters and contractions such as *'s*, *'t*, and *'d*.

  At this point we have taken into account that the IRS must return articles with a publication date previous to the article we are reading. This will mean articles that have an earlier time stamp of "published_date".

  Figure 2 shows an example of the creation of the query with ID 809 using only the title of the article and taking into account the restriction of the publication date.

- **Information Retrieval System**

  TREC proposes to use trec_eval (NIST, 2017). The trec_eval is a tool used to evaluate rankings, either documents or any other information that is sorted by relevance. This system return us the results using a file that contains a ranking of documents for each query automatically generated by the application.

  Our IRS will return 1000 relevant documents for each query and then we will remove from the list of relevant documents the documents with the same ID as the input query.

- **TF-IDF Vectors**

  We can represent each document obtained from the IRS as mutually comparable vectors of terms. These vectors are used to compare documents by similarity and put them into clusters.

  A good approach is to assign a weighting to each term in the document and put that weighting in the vector with TD-IDF(Ramos and others, 2003). TF-IDF (Term Frequency-Inverse Document Frequency) is a very common algorithm to transform text into a meaningful representation of numbers. The technique is widely used to extract features across various Natural Language Processing (NLP) applications. With TF we normalize the occurrence of each word with the corpus size, IDF measures the importance of a specific term by its relevance within the document excluding English stop words.

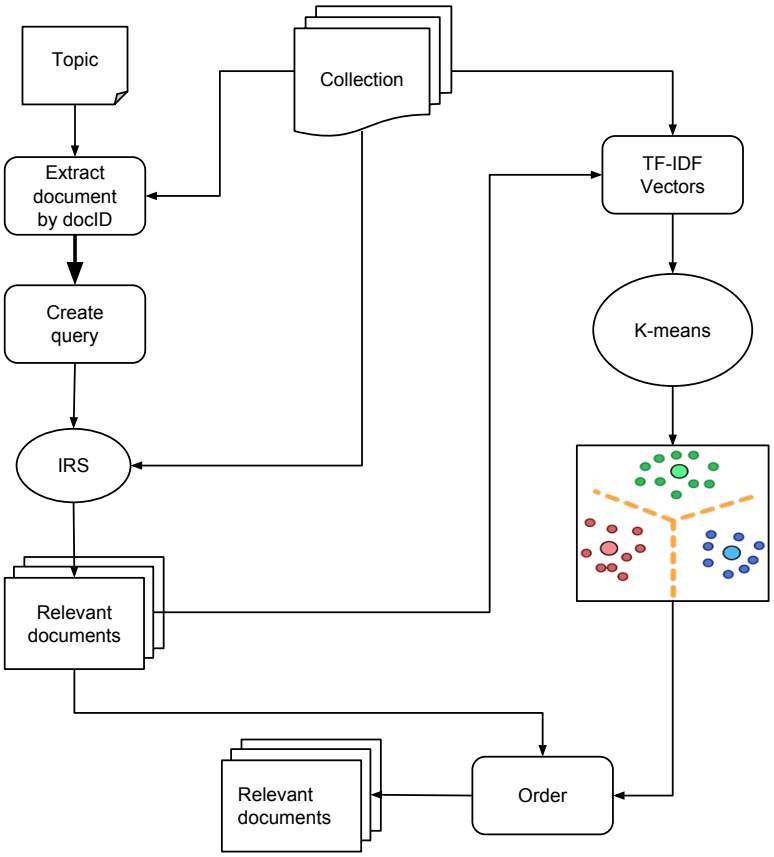

Figure 1: Architecture of the approach

To perform the TF-IDF of each relevant document returned by the IRS we will take into account the *title* and *sanitized_html*.

- **K-means**

  The method chosen for grouping documents is K-means. K-means is one of the most popular clustering algorithms(Jain et al., 1999). K-means stores k centroids that it uses to define clusters. A point is considered to be in a particular cluster if it is closer to that cluster's centroid than any other centroid.

  To implement this step we use the Python library *sklearn.cluster.kMeans*, with k=10, the default initialization method k-means++, max_iter=1000 and the number of time the K-means algorithm will be run with different centroid seeds (n_init) by default equal to 10 (Pena et al., 1999).

- **Order**

  Finally, the K-means algorithm will return 10 clusters.

  Each cluster contains a list of documents, we will take the first 10 documents of each cluster and to finish, this list of 100 documents (10 clusters x 10 first documents) is sorted according to its relevance obtained from the ISR results file. If one or more clusters do not have 10 documents, the final file will contain less than 100 relevant documents.

4 Results

In following Table 1 we show the results we have obtained in our experiments.

At this moment we do not have the results of all participants, but the organization has provided us for each measure and topic: the minimum, the average and the maximum.

A total of 23 Runs have been sent for this task and we have calculated the average obtained

```
<query>
  <type>indri</type>
  <number>809</number>
  <text> #combine(Europe will send a rover to Mars but won protect
  Earth from an asteroid #less(published_date 1480958455000))</text>
</query>
```

Figure 2: Example of query 809 with title

Run	***NDCG_cut_5***	***P_5***	***Recall_5***	***RPrecision***
Baseline. Abstract	0.2989	0.6120	0.0356	0.2203
Baseline. Title	0.2404	0.5280	0.0289	0.1975
Baseline. Title + Abstract	0.2986	0.6120	0.0356	**0.2291**
Clustering. Abstract	0.2989	0.6120	0.0356	0.0962
Clustering. Title	0.2404	0.5280	0.0289	0.1331
Clustering. Title + Abstract	0.2986	0.6120	0.0356	0.1053

Table 1: Results obtained.

for each measurement used. The measures have been 0.27665, 0.56, 0.030226 and 0.128234 for NDCG_cut_5, P_5, Recall_5 and RPrecision respectively.

We check that for the measures NDCG_cut_5, P_5 and Recall_5 we improve the average in all cases except when we use only the titles as input query. On the other hand, for the RPrecision measure we have lower results if we use the clustering algorithm.

Finally, we can see that the best system is the baseline using titles and abstracts in the query. We improve the average of all the participants by 78% in the RPrecision measure.

5 Conclusion

In this article our group presents its first participation in the task of CLEF News. We have made a first approach clustering and TF-IDF for word processing. We can coclude that clustering is not a goog approach because we do not overcome the Baseline . We must analyse the relevant files and see where our systems fail.

We will continue grouping the different news by subject to offer the user different perspectives of a news item. We will need to analyze each piece of news carefully for greater accuracy. For this, we will use NLP techniques and tools for word processing as present Bouras in (Bouras and Tsogkas, 2010) using Wordnet.

Acknowledgments

This work has been partially supported by a grant from Fondo Europeo de Desarrollo Regional (FEDER) and REDES project (TIN2015-65136-C2-1-R) from the Spanish Government.

References

Christos Bouras and Vassilis Tsogkas. 2010. W-kmeans: clustering news articles using wordnet. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 379–388. Springer.

Bruce Croft and Jamie Callan. 2016. The Lemur Project. https://www.lemurproject.org/.

Shudong Huang, Donna Harman, and Ian Soboroff. 2018. The News Track. http://trec-news.org/.

A. K. Jain, M. N. Murty, and P. J. Flynn. 1999. Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323, September.

Amy Mitchell, Elisa Shearer, Jeffrey Gottfried, and Michael Barthel. 2016. The modern news consumer: News attitudes and practices in the digital era. http://www.journalism.org/2016/07/07/the-modern-news-consumer/.

NIST. 2017. TREC eval software. https://trec.nist.gov/trec_eval/.

NIST. 2018. TREC Washington Post Corpus. https://trec.nist.gov/data/wapost/.

José M Pena, Jose Antonio Lozano, and Pedro Larranaga. 1999. An empirical comparison of four initialization methods for the k-means algorithm. *Pattern recognition letters*, 20(10):1027–1040.

Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 133–142.